

On Computing Mapping of 3D Objects: A Survey

XIN LI, Louisiana State University

S. S. IYENGAR, Florida International University

We review the computation of 3D geometric data mapping, which establishes one-to-one correspondence between or among spatial/spatiotemporal objects. Effective mapping benefits many scientific and engineering tasks that involve the modeling and processing of correlated geometric or image data. We model mapping computation as an optimization problem with certain geometric constraints and go through its general solving pipeline. Different mapping algorithms are discussed and compared according to their formulations of objective functions, constraints, and optimization strategies.

Categories and Subject Descriptors: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling

General Terms: Algorithms

Additional Key Words and Phrases: 3D Geometric mapping, shape matching and registration

ACM Reference Format:

Xin Li and S. S. Iyengar. 2014. On computing mapping of 3D objects: A survey. *ACM Comput. Surv.* 47, 2, Article 34 (December 2014), 45 pages.

DOI: <http://dx.doi.org/10.1145/2668020>

1. INTRODUCTION

In recent years, the advancement of 3D acquisition technology has greatly enriched the availability of digital geometric data. A fundamental problem that exists in many data processing tasks is to correlate these 3D raw acquisitions, so that the analysis, comparison, and integration of different objects or a same object under different modalities can be done in a unified coordinate system or parametric domain. This article studies the problem of establishing a bijective map or correspondence between two given geometric shapes, which is simply referred to as *shape mapping* in the following.

Specifically, given two 3D objects $M_0 \subset \mathcal{R}^3$ and $M_1 \subset \mathcal{R}^3$, the goal of shape mapping is to solve a map $f : M_0 \rightarrow M_1$ between them following some *geometric* and *semantic* criteria. Desirable *geometric properties* of a mapping function include the preservation of angles, area/volume, distances, or other metrics under the transformation induced by the mapping. Desirable *semantic properties* of a mapping include the modeling/alignment of various functional, material, or anatomical features/structures/layers of the data.

This work is partially supported by the National Science Foundation IIS-1320959, IIS-1251095, and CNS-1158701; Louisiana Board of Regents LEQSF(2009-12)-RD-A-06, LEQSF-EPS(2009)-PFUND-133, and LEQSF-EPS(2013)-PFUND-312; and the National Natural Science Foundation of China No. 61170323.

Authors' address: X. Li, School of Electrical Engineering and Computer Science, and Center for Computation and Technology, LSU, Room 2002, Louisiana Digital Media Center, 340 E Parker Blvd, Baton Rouge, LA 70808, USA; S. S. Iyengar, School of Computing and Information Sciences, Room 351, Computing and Information Sciences (ECS) building, Florida International University MMC Campus, Miami, FL 33199, USA.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© 2014 ACM 0360-0300/2014/12-ART34 \$15.00

DOI: <http://dx.doi.org/10.1145/2668020>

The map $f : M_0 \rightarrow M_1$ can usually be represented as scalar fields f_x, f_y, f_z defined on M_0 , which minimize an objective function $E(f, M_0, M_1)$ measuring geometric and/or semantic distortions. We classify and compare different mapping algorithms by analyzing their optimization models and solving strategies.

In some scenarios, instead of solving the mapping function f in 3D space (in which the objects embed) explicitly, a convenient implicit way to obtain f is using the *parametric representations* of M_0 and M_1 on a common domain Ω . A *parametric representation* or a *parameterization* of M_i on Ω can be defined as a bijective map $\phi_i : \Omega \rightarrow M_i$ from a canonical domain Ω to the 3D shape. With the parameterizations of both shapes onto a same domain, the shape mapping $f : M_0 \rightarrow M_1$ can be obtained by the composition $f = \phi_1 \circ \phi_0^{-1}$. For example, to map two surface patches M_0 and M_1 that are topological disks, one can first parameterize both M_0 and M_1 onto a unit planar disk $D : \{(x, y) | x^2 + y^2 \leq 1\}$ via $\phi_0 = D \rightarrow M_0$ and $\phi_1 = D \rightarrow M_1$, then obtain the bijective correspondence between M_0 and M_1 by the composition: $\forall p \in M_0, f(p) = \phi_1 \circ \phi_0^{-1}(p) \in M_1$.

The *shape mapping* problem can be generalized to correlate multiple objects. For $n + 1$ objects M_0, M_1, \dots, M_n , one can either compute n explicit maps $f_i : M_0 \rightarrow M_i, i = 1, \dots, n$, or solve $n + 1$ implicit parameterizations, $\phi_i : \Omega \rightarrow M_i, i = 0, 1, \dots, n$, then respectively, compose interobject correspondence $f_{i,j} : M_i \rightarrow M_j$ using either $f_{i,j} = f_j \circ f_i^{-1}$ or $f_{i,j} = \phi_j \circ \phi_i^{-1}$.

Geometric mapping has many applications in tasks involving spatial or spatiotemporal alignment or correlation problems in graphics, vision, medical imaging, computer-aided design, and other scientific/engineering fields. A few examples are illustrated in Figure 1: in computer vision and reverse engineering, shape mapping is needed for digitization of 3D objects during the scan reconstruction stage, as geometric objects often need to be scanned from multiple angles and the overlapped regions from different scans are mapped and then stitched together to compose the complete model (a). In forensics and archeology, effective mapping among fragmented pieces can be used to reassemble and restore damaged geometric objects (b). In graphics and animation, shape mapping can be used to generate a shape-interpolation that animates the morphing from one object to another (c). In medical image analysis, sequential CT/MRI scans can be correlated to model the deformation of the organs for motion tracking and prediction (d).

1.1. Classification of Cross-Shape Mapping Algorithms

In general, mapping computation can be formulated as an optimization problem. We classify mapping algorithms through the way the (a) *objective functions* and (b) *constraints* are formulated, and (c) *optimization strategies* are adopted in problem solving. The classification is elaborated here and illustrated in Table I.

—First, mapping computation can be modeled using different objective functions.

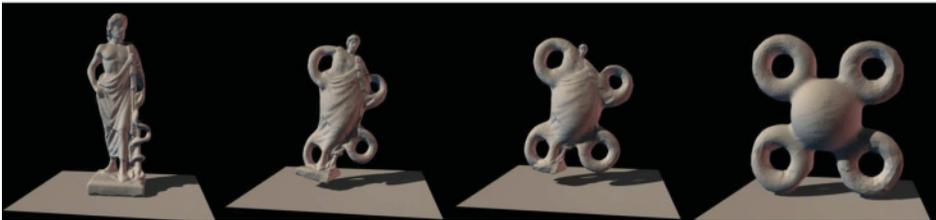
- (1) Using the *dimensionality* of the shape, geometric mapping can be classified as curve (1D-manifold) mapping, surface (2D-manifold) mapping, volumetric (3D-manifold) mapping, and so forth. If the input is a sequence of shapes, we call the mapping function that correlates all these shapes as dynamic (n Dimensional plus Temporal, or $nD + T$) shape mappings. This article focuses on the mapping of 2D, 3D, and 2D/3D + T data, whose modeling and analysis applications most widely exists in scientific tasks.
- (2) Based on its *completeness*, mappings can be *complete* or *partial*. A *complete mapping* maps an entire object bijectively onto the other. In contrast, a *partial mapping* only correlates some commonly matched subregions from different shapes. For example, in 3D damaged geometric data restoration (Figure 1(b)), fragmented



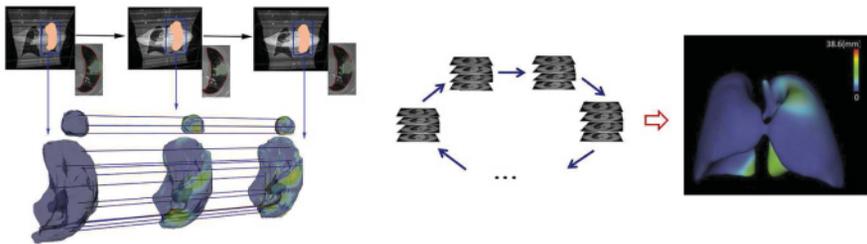
(a) Reconstruction in computer vision and reverse engineering: 3D scans of an object from different angles are mapped and stitched together to compose the complete geometry.



(b) Damaged geometric object restoration in forensics and archaeology [Li et al. 2011; Yin et al. 2011].



(c) Cross-shape interpolation or morphing in computer graphics and animation [Lee et al. 1999; Li et al. 2008a]. The Greek sculpture model is provided courtesy of IMATI by the AIMSHAPE-VISIONAIR Shape Repository.



(d) Motion modeling and trajectory prediction of deforming organs in medical image analysis using spatial-temporal mapping [Iyengar et al. 2012; Xu and Li 2013b].

Fig. 1. Several applications of different types of shape mapping.

pieces share only partial geometric similarities with adjacent pieces, and partial mapping between these overlapping fracture regions is needed to perform correct reassembly and completion. In contrast, in morphing animation (Figure 1(c)), a complete map between two entire given models is needed to construct a natural interpolation between these shapes.

- (3) The *transformation model* is another factor that dictates the formulation of objective functions in mapping computation. In some tasks, meaningful mappings

Table I. Classifications of Cross-Shape Mapping Schemes and This Article's Organization

Objective Functions	Shape dimensions	Mapping nD manifolds; Mapping $nD + T$ manifolds
	Representation schemes	Parametric vs. nonparametric representations, explicit vs. implicit representations
	Transformation models	Rigid mapping vs. nonrigid mapping
	Distortion	Geometric deviation; metric distortion
	Completeness	Complete mapping vs. partial mapping
Constraints	Feature alignment	With vs. without feature correspondence
	Other constraints	Bijectivity, smoothness, regularization
Optimization Methods	Optimization strategies	Deterministic optimizations, stochastic optimization

are restricted to simple global transformations that preserve Euclidean geometric properties, thus the maps to be solved are simply translations, rotations, or their combinations. For example, in 3D scan reconstruction (Figure 1(a)), multiple scans of the same object differ only by coordinate transformations, thus one only needs to find suitable *rigid transformations* for their stitching. In some other scenarios, f can be any free-form transformation where the deformation can be locally different in a free manner. For example, to track the tissue motion from dynamic medical images (Figure 1(d)), the soft tissues and organs are deforming nonrigidly. Therefore, one needs to seek a good nonrigid deformation that minimizes some predetermined geometric and biophysical energies.

- Second, considering different constraints to be enforced in the optimization, we can classify mapping computation by using different shape representation schemes, by whether or not certain features are incorporated, or by either representing the map function explicitly or implicitly.
- Finally, optimization problems can be solved using different strategies according to the structure of the objective functions and constraints. We classify them as deterministic and stochastic optimization approaches, which are suitable for solving maps in different scenarios.

1.2. Comparison with Existing Surveys

Shape mapping is related to 2D image registration in computer vision, and 3D geometric parameterization and correspondence in graphics. There are a few surveys in these related fields. However, the problem of shape mapping computation discussed in this article, which aims to establish a bijective map of given geometric data that minimizes metric or semantic distortions, is different from those problems.

2D image registration, studied in computer vision and medical imaging, seeks a transformation that maps the spatial coordinates (grid pixels) on one image to those on the second image [Brown 1992; Pluim and Fitzpatrick 2003]. 3D cross-shape mapping has several fundamental differences from 2D image registration, and many image-based methods cannot be generalized for mapping 3D shapes. (1) First, due to the fundamental principal of imaging, the 3D objects are projected to 2D images, losing depth information. Under the projective transformations, most Euclidean or Riemannian geometric properties of 3D shapes—such as their structure and length—are not preserved. Therefore, instead of preserving intrinsic geometric properties, image registration algorithms often seek projective invariants, affine invariants, or illumination invariants. (2) Also, *image registration* algorithms are usually formulated as finding the transformation between the two image spaces. The solved transformation is extrinsic and deforms the space rather than the object itself. Shape mapping, in contrast, seeks

an intrinsic deformation of the shape itself. This transformation is determined by the geometric properties (such as its topology, geometric metrics) of the object, rather than how the object embeds in the 3D space. Therefore, unlike the image registration—that has simpler topological structure and whose optimization can be formulated/solved in the Euclidean spaces directly—a mapping between 3D shapes is often defined on a curved geometry/manifold, and its optimization can have more complicated constraints and is more expensive to compute. (3) Finally, 3D shapes possess complete geometric shape information, but many do not have colors and textures, thus the modeling of features on images and on 3D shapes is often quite different.

3D shape alignment or registration is another closely related problem to shape mapping. An alignment or registration seeks a best transformation (within a certain model space) to lay a source object onto the target object. Achieving such a transformation is not always equivalent to computing a mapping between two shapes. For example, when the goal is to find a global rigid alignment between two given objects, the transformation is not required (and usually, not possible) to bijectively map the source object M_1 onto the target object M_2 . An alignment T , designed to reduce alignment error, such as the Hausdorff distance, often does not introduce a map among points of M_1 and M_2 . Tam et al. [2013] focused on discussing different registration algorithms for the data fitting application, which can be done through aligning a clean and complete template to the raw/incomplete data. For applications like this, establishing a one-to-one map is often not necessary.

The *shape correspondence* problem attempts to establish either coarse or dense correspondences between 3D shapes. Coarse correspondence focuses on mapping a small set of feature points on two objects, while dense correspondence aims to map more densely sampled point sets on the objects. Kaick et al. [2011] conducted a survey on this topic. Research on the shape correspondence problem mainly focuses on matching semantically correlated points or regions, and the shapes (or parts of the shapes) are often assumed to be either perfectly or nearly isometric to each other so that their global intrinsic structures are relatively stable. Major applications of this research include shape recognition/retrieval, shape segmentation, and time-varying reconstruction [Sahillioglu 2012]. The criteria for effective correspondence are often the successful semantic alignment in various domain-specific applications. Therefore, Kaick et al. [2011] focused on feature point description and matching; although regularization is also considered in shape correspondence computation, the bijectivity and intrinsic angle/volume distortion is often not thoroughly examined.

The *surface* or volume parameterization problem, which finds a bijective map between a given surface/volume and a suitable parameter domain, is another closely related topic in graphics and geometric processing. Most existing parameterization surveys [Floater and Hormann 2005; Sheffer et al. 2006; Hormann et al. 2007] discuss the flattening of 3D shapes onto a planar region. Effective parameterization often provides a useful computational tool for implicit cross-shape mapping computation. However, shape parameterization does not directly achieve intershape mapping if the shapes have nontrivial topology. The parameterization of surfaces or volumes with nontrivial topologies inevitably contain singularities whose numbers and distributions may not be easily controllable, and without consistently distributed singularities, two parameterizations cannot be used to compose a valid intershape mapping.

1.3. Organization

We first introduce the necessary background and terminologies in Section 1.4. Then following Table I, we will elaborate mapping computation algorithms classified by different objective functions, constraints, and optimization methods, respectively, in Sections 2, 3, and 4.

1.4. Notations and Problem Formulation

Smooth geometries or manifolds are often discretized to polygonal meshes for storage and computation. 3D surfaces (solids) can be triangulated (tetrahedralized) and adaptively refined to approximate the original geometry. The generalized terminology for geometric primitives—such as points, edges, triangles, and tetrahedra—are referred to as *simplices*; a triangular or tetrahedral mesh is defined as a *simplicial complex*.

Suppose $k + 1$ points $\{v_0, v_1, \dots, v_k\}$ are affinely independent in \mathcal{R}^n , $n \geq k + 1$; the k -dimensional simplex (also denoted as k -simplex in the following) $[v_0, v_1, \dots, v_k]$ is the minimal convex set including all of them in other words,

$$\sigma = [v_0, v_1, \dots, v_k] = \left\{ x \in \mathcal{R}^n \mid x = \sum_{i=0}^k \lambda_i v_i, \sum_{i=0}^k \lambda_i = 1, \lambda_i \geq 0 \right\}. \quad (1)$$

We call $\{v_i\}$ the *vertices* of the simplex σ , and all corresponding $\{\lambda_i\}$ the *barycentric coordinates* of point $x \in \sigma$. Following this definition, the points, edges, triangles, and tetrahedra are 0, 1, 2, and 3-simplices, respectively. If σ is a simplex and $\tau \subset \sigma$ is also a simplex, then τ is called a *facet* of σ , denoted as $\tau \leq \sigma$. *Simplices* can be coherently glued together to form *simplicial complexes*. A *simplicial complex* M is the collection of facets of a finite number of simplices, any two of which either are disjoint or share a common facet, that is, (1) if a simplex σ belongs to M , then all its facets $\tau \leq \sigma$ also belong to M ; and (2) if $\sigma_1, \sigma_2 \in M$, then $\sigma_1 \cap \sigma_2 = \emptyset$ or $\sigma_1 \cap \sigma_2 \leq \sigma_1, \sigma_2$. The dimension of a complex M is the highest dimension of any simplex $\sigma \in M$. If M 's dimension is k , we call it a k -complex. Triangular meshes and tetrahedral meshes are 2-complexes and 3-complexes, respectively.

A *piecewise linear mapping* function f between two n -complexes M_1 and M_2 is defined on vertices of M_1 , namely, $f(v)$ maps $v \in M_1$ to a point x in an n -simplex of M_2 , and x 's location is indicated by the $(n + 1)$ -dimension barycentric coordinates (see Equation (1)). The mapping function $f : M_1 \rightarrow M_2$ between two shapes in 3D can be explicitly described by three piecewise linear scalar fields (ϕ^1, ϕ^2, ϕ^3) defined on vertices of M_1 . These piecewise linear scalar fields have constant gradients $(\nabla\phi^1, \nabla\phi^2, \nabla\phi^3)$ within each n -simplex of M_1 .

2. OBJECTIVE FUNCTIONS

Shape mapping algorithms can be classified by the different objective functions to be optimized. We discuss different mapping algorithms from four major aspects in Sections 2.1 through 2.4.

2.1. Dimensionality and Representations of Shapes

Shape Dimensions. We can classify the mappings between/among shapes using the dimensionality of the shapes. Curves, surfaces, and solids can be discretized as $1D$, $2D$, and $3D$ simplicial complexes. Higher dimensionality of the shapes increases the complexity of topology and geometry of the data and their mapping computation.

- Curve mapping has been studied [Sebastian et al. 2003; Pajdla and Gool 1995] and applied in computer vision for handwriting or signature recognition [Meenakshi et al. 2004], 2D shape contour recognition [Ferrari et al. 2008], and in computer graphics for skeleton matching (which can further guide cross-surface [Wang et al. 2009; Li et al. 2006; Li 2008] or cross-volume [Li et al. 2010] mapping) or shape retrieval [Sundar et al. 2003; Hilaga et al. 2001]). Curve mapping, finding most of its applications in image analysis, is not the focus of this survey; we refer the readers to Sebastian et al. [2003] and Glaunes et al. [2008] for more detailed discussion on its computation.

- The computation of mapping between two surfaces, deeply rooted in Riemannian geometry and differential geometry, has been studied in the past decade due to its broad scientific applications. We will elaborate on intersurface mapping algorithms in the following sections. Intersurface mapping between general 2D manifolds has many applications, such as morphing [Lee et al. 1999; Kraevoy and Sheffer 2004; Schreiner et al. 2004; Li et al. 2008b] and texture mapping/transfer [Li et al. 2008a] in graphics, data fitting and completion [Kraevoy and Sheffer 2005], detail transfer [Biermann et al. 2002] and surface editing [Lévy 2003] in geometric modeling and processing; mesh generation [Alliez et al. 2003; Wang et al. 2009] and spline construction [Wang et al. 2008b, 2012; Cao et al. 2009, 2012] in computer-aided geometric design, 3D scan reconstruction and object tracking [Wand et al. 2007; Wang et al. 2008a] in computer vision and medical image analysis.
- The mapping of solid models has attracted great attention in the past few years. Solid objects have richer contents than their boundary surfaces, especially if material, intensity, or other structural information of the data should be considered. Volumetric mapping that correlates geometric solid data has broad applications in physics-based simulation [Ju et al. 2005; Joshi et al. 2007a], fluid dynamics, material modeling, trivariate spline construction [Martin et al. 2008; Wang et al. 2012; Li et al. 2013] and mesh generation [Li et al. 2007; He et al. 2009a; Gregson et al. 2011; Yu et al. 2013; Nieser et al. 2011; Li et al. 2012], medical data fusion and difference analysis [Wang et al. 2004], and medical tracking [Metz et al. 2011; Xu et al. 2012; Xu and Li 2013a].

Pairwise and Groupwise Mappings. We can classify mapping problems as the *pairwise* mapping between two objects and the *groupwise* mapping among multiple objects. In some tasks, groupwise mapping reduces to solving many pairwise mappings. One example is shape comparison: given objects M_1, \dots, M_k and a query subject N , to find N 's most similar object(s) by matching N with each M_j then report the one(s) with the smallest difference [Sundar et al. 2003]. Groupwise mapping computation in this case can reduce to k pairwise mappings. In some other scenarios, groupwise mapping is used to correlate a temporal sequence of deforming objects. Then the coherence of the deformation in the time dimension becomes an extra guidance or constraint in groupwise mapping computation, and the resultant $nD + T$ mapping correlates the entire sequence of deforming n -dimensional manifolds. Its computation then can be done on an $n + 1$ dimensional space to allow the extra constraint on the time dimension.

Shape Representations. The formulation of shape mapping is dictated by the way that shapes are represented. The shape representation schemes can be classified into two main classes: *parametric* representations and *implicit* representations. A *parametric representation*, such as the polygonal mesh or splines, defines a k -D manifold shape M by a vector-valued parameterization function $\phi : \Omega \rightarrow M$, where the parameter domain Ω is in \mathcal{R}^k ($k = 1, 2, 3$ for curves, surfaces, and solids, respectively). An *implicit representation* defines a shape as the zero set of a scalar-valued function $\varphi : \mathcal{R}^n \rightarrow \mathcal{R}$, that is, $M = \{\mathbf{x} \in \mathcal{R}^n | \varphi(\mathbf{x}) = 0\}$. Different representations have their own pros and cons in modeling different geometric saliency. According to different mapping applications, choosing desirable shape representation schemes is important. We discuss a few commonly used representation schemes in mapping computation.

—*Polygonal Meshes.* A widely used 3D parametric shape representation in graphics is the polygonal (especially, triangular) mesh. Curves, surfaces, and solids can be adaptively approximated using piecewise linear simplicial complexes as accurately as required. Using this representation, mapping computation is often performed using finite element methods [Schreiner et al. 2004; Kraevoy and Sheffer 2004; Li et al. 2008a].

- Spline and Subdivision Representations.* The tensor-product spline representation [Farin 2002] is the standard representation in current CAD systems. Described using piecewise polynomial or rational basis functions, splines can be considered as a generalization of piecewise linear mesh representations to the piecewise polynomial/rational scheme. The subdivision representation, which can be considered as splines with singularities, represents shapes by repeated refinement of control meshes, where positions of vertices are adjusted following certain local averaging rules [Zorin et al. 2000]. In mapping computation, if the input shapes are given in spline or subdivision representations, mapping optimization can be formulated as solving the control points or the transformation of the control points [Donato and Belongie 2002].
- Implicit or Volumetric Representations.* Widely used in data completion, the implicit representations [Osher and Fedkiw 2002] describe shapes using a scalar function to define a shape by classifying each point in the space to either inside, outside, or on the shape. Common implicit representations include level-set functions, algebraic surfaces, and discrete voxelization. In mapping computation, implicit representations have several advantages: flexible description of shapes with multi-component or complicated/changing topology, natural prevention of self-intersections, and so forth. But a limitation of implicit representations is the inaccuracy and costliness in representing and preserving fine geometric details and intrinsic metrics.
- Feature-Based Representations.* Shape representation defined on *features* of the 3D shape are sometimes effective in shape mapping. For example, *Medial Representation* [Siddiqi and Pizer 2009] approximates a shape using radius functions defined on the medial axis of the shape. This medial axis, together with the associated radius function, reduce the dimensionality and complexity of the representation of the original shape, and is suitable for efficient shape matching. The skeleton can provide global guidance in nonrigid shape matching [Vlasic et al. 2008]. *Generalized Cones and Generalized Cylinders (GC)* representations approximate elongated shapes using a group of cylinders defined on a skeletal graph. The GC representation can be used in shape mapping due to its efficient representation of intra-subpart relationship [Chuang et al. 2004]. It is, however, sometimes insufficient in representing general free-form shapes with convoluted details.
- Spectrum-Based Representations.* Spectral representations, such as Laplace-spectra [Reuter et al. 2005], spherical harmonics [Kazhdan et al. 2003; Huang et al. 2005], manifold harmonics [Vallet and Lévy 2008], Zernike [Novotni and Klein 2003], often naturally offer a multiresolutional description of the shape that is often desirable in improving the efficiency and reliability of mapping computation.

2.2. Transformation Model

By choosing different models, one shape can undergo different transformations to match the second object. This determines the search space and the “softness” of the mapping. Based on the transformation model, mapping models can be classified as *globally uniform* transformations and *globally nonuniform* transformations. We examine the rigidity of an algorithm through the degree-of-freedom (DOF) allowed in describing the map f :

- Mapping algorithms solving a small (constant) number of DOF for a global transformation (e.g., global rigid/affine transformations, global free-form deformation [FFD] [Sederberg and Parry 1986]) are classified as globally uniform transformations.
- Mapping algorithms allowing different subregions to deform differently (thus usually make the DOF proportional to the mesh element size) are classified as globally nonuniform transformations.

2.2.1. Globally Uniform Transformations. Global rigid transformations and other low-dimensional transformations [Li and Guskov 2005; Gelfand et al. 2005] are often used in aligning objects that undergo no self-deformations. Given general shapes M_1 and M_2 , solving $f : M_1 \rightarrow M_2$ with the restriction that f be rigid often leads to no solution. But finding partial mapping to rigidly align and merge two shapes is important in 3D scan reconstruction, geometric fragments reassembly, moving object tracking, and the like. In addition, in the computation of optimal nonrigid cross-shape mapping, effective rigid mapping can be used as a prealignment or initial estimate, which can greatly improve the efficiency and robustness of the subsequent optimization of finer mapping.

Under a *global rigid transformation*, a shape M_1 undergoes rotations and translations to align with M_2 , and the entire transformation can be represented using a 3×3 rotation matrix \mathbf{R} and a 3×1 translation vector \mathbf{T} . The rotation \mathbf{R} usually has 3 independent degrees of freedom (rotation angles about the x-, y-, and z- axes); the translation has 3 as well. The solving of rigid function $f : M \rightarrow N$ can be decomposed into two problems: (1) finding the correspondence; and (2) finding the best transformation $f = (\mathbf{R}, \mathbf{T})$ that matches the corresponded elements, where \mathbf{R} and \mathbf{T} represent the associated rotation and translation. Solving each subproblem with the other fixed is much easier than solving either one without information regarding the other.

A class of rigid registration methods solves the second problem without knowing the correspondence. For example, one can normalize the mass center and principal axes to place each 3D shape canonically in \mathcal{R}^3 . The mass center or principal axes can be computed by moment analysis or through eigenvectors of the covariance matrix of the shape [Bronstein et al. 2008]. Other related algorithms in this category include the generalized Hough Transform [Ballard 1981; Hecker and Bolle 1994] and geometric hashing [Hecker and Bolle 1994]. Although these methods work well for solving rigid transformation that has a small DOF, without knowing the correspondence, they cannot be easily generalized to solve effective nonrigid transformation between arbitrary shapes. [Reyes et al. 2007] present an effective noniterative algorithm, which combines the power of expression of geometric algebra with the robustness of tensor voting, to rigidly correspond two 3D point sets.

If the point-to-point correspondence is known, solving the optimal rigid transformation is easy. One can minimize the following least squares with 6 unknown variables:

$$\epsilon = \sum_{p_i \in M_1, q_i \in M_2} \|q_i - \mathbf{R}p_i - \mathbf{T}\|^2, \quad (2)$$

where $\{p_i\}$ and $\{q_i\}$ are the corresponding points. It is possible to develop a closed-form solution for this problem. [Eggert et al. 1997] compared four classic rigid transformation computation algorithms [Arun et al. 1987; Horn 1987; Horn et al. 1988; Walker et al. 1991] that solve optimal rigid transformations (with given shape correspondence) using singular value decomposition or eigensystem computation.

A more general approach is to solve the transformation and correspondence simultaneously. A widely used approach in registration is the Iterative Closest Point (ICP) algorithm [Besl and McKay 1992], which is an iterative procedure that assigns correspondence using the nearest neighboring information and refines the transformation using this correspondence information. The ICP algorithm converges quickly to a local minimum. The initial guess is important and geometric noise can affect the registration result. A lot of variant ICP algorithms [Amberg et al. 2007] have been developed to make it robust against initial guess and noise.

Another common low-dimensional model is the *affine transformation*. This model allows that the deformation consists of not only translation and rotation (as rigid-motion), but also scaling and shearing. It is often used for registering objects with

gross-overall distortion, such as the registration of MRI data to CT data. More typically, affine transformation can be used as an approximation for fully nonrigid transformation. Two other representative low-dimensional representation transformations are the *bilinear* and *projective transformations*, mostly used in image registration. With homogeneous coordinates, all these four low-dimensional transformations can be represented as 4×4 matrices.

2.2.2. Globally Nonuniform Transformations. When different objects are being matched or when free-form deformations are substantial, one needs to solve nonrigid transformations when matching different shapes [Brown and Rusinkiewicz 2007; Pauly et al. 2005; Sumner and Popović 2004; Li et al. 2008]. Nonrigid transformations are also considered in some 3D scan reconstruction, as sometimes even very expensive 3D scanners suffer from calibration problems that need local alignment. Nonrigid transformations are usually nonlinear, namely, there is no simple matrix representation for them. In such a case, solving too many degrees of freedom for local transformations may lead to undesirable results. Therefore, *geometric distortion* and extra *smoothness* or *regularity* constraints to restrict the nonrigidity should be incorporated in the search of desirable cross-shape mapping. These distortions, smoothness, and regularity terms, considered in many mapping optimization algorithms, will be discussed in the following sections.

2.3. Distortion Measurement

Mapping quality can be classified based on the different distortion metrics being used, which is either extrinsic (Section 2.3.1) or intrinsic (Section 2.3.2).

2.3.1. Extrinsic Mapping. Extrinsic mapping algorithms solve a deformation of the source shape M_1 toward the target shape M_2 . The deformation is usually represented using a 3D vector function, often referred to as displacement field, defined on the spatial domain Ω where $M_1 \subset \Omega \subset \mathcal{R}^3$. Physical or geometric energies are usually used to govern the transformation in mapping computation so that the deformation is spatially smooth or physically natural. Widely used energies include elastic strain energy [Joshi et al. 2007b], biharmonic energy or thin-plate splines [Bookstein 1989], viscous fluid model [Christensen et al. 1996], diffeomorphic metric model [Glaunes et al. 2004], and radial basis functions [Fornet et al. 2001]. Extrinsic mapping usually indicates a physically intuitive transformation, and its computation reduces to iterative optimizations. One common strategy is to first compute a (weighted) correspondence set in which each correspondence has a confidence value. These correspondence pairs and their confidence values are used to remove the outliers. M_1 and M_2 are first rigidly aligned, then M_1 is nonrigidly deformed to match M_2 . When the given shapes are similar, that is, the deformation is near identity, the extrinsic mapping methods are usually efficient. Therefore, they are widely applied in registering dynamically deforming surface sequences [Wand et al. 2007; Pekelny and Gotsman 2008; Wand et al. 2009; Li et al. 2009; Allen et al. 2003; Zhang 1994; Rusinkiewicz and Levoy 2001], in which case one can assume that the adjacent frames do not change drastically. However, when the geometric shapes have complicated convolutions or differ by a large deformation, an extrinsic mapping algorithm could fail to produce correct correspondence and easily get trapped in local minima.

2.3.2. Intrinsic Mapping. When the shapes to map are very different or undergo large deformation, an alternative mapping strategy is to compose the parameterizations of these shapes on a common domain while minimizing the intrinsic metric distortion of the composed mapping function. Two general approaches are adopted in cross-shape mapping: the parameterizations onto a low-dimensional domain and parameterizations onto high-dimensional space.

- A surface is intrinsically a $2D$ manifold and can be parameterized onto a $2D$ planar region, upon which the representation of each surface point needs only a $2D$ coordinate. In the optimization of surface mapping, this reduces the DOF on each vertex from an explicit $3D$ coordinate to $2D$, and it avoids the nonlinear geometric constraints that enforce the image of each source surface point to be on the target surface. Two comprehensive surveys about surface parameterization are given in Floater and Hormann [2005], and Sheffer et al. [2006].
- Another intrinsic approach maps shapes onto a higher-dimensional space. The idea is to embed the shapes in a space in which the registration can be better approximated as (near-) rigid alignments. To create an effective intrinsic embedding, one needs to find an intrinsic representation of the shape invariant to extrinsic deformations, so that the intrinsic geometry can be encoded using the extrinsic coordinates in the new metric space in which we embed the shapes. The multidimensional scaling [Elad and Kimmel 2003; Bronstein et al. 2008] and the spectral transform [Jain et al. 2007; Mateus et al. 2008; Sahillioglu and Yemez 2010] are two popular strategies to construct the embedding. A comprehensive survey on this topic is given in Zhang et al. [2010].

Parametric representations preserving intrinsic geometric metrics are desirable in composing the cross-shape mapping function. We can classify intrinsic mapping/parameterization methods via distortions that they aim to eliminate/minimize: *distance-preserving*, *angle-preserving*, and *area/volume-preserving* maps. These three general types of maps are referred to as the isometry, conformal mapping, and equiareal mapping. We start with the conformal mapping, as most effective surface mappings should restrict the distortion of the local shape, that is, angles between intersecting curves, and usually incorporate conformality as an important criterion. Conformal mappings have many desirable properties due to their connection to the Riemannian geometry and complex function theory. We briefly review the mathematic background for Riemann surface, conformal and harmonic mappings, and refer interested readers to Gu and Yau [2007] for more details.

Planar Conformal Mapping. Suppose $f : D \rightarrow \mathbb{R}$ is a real valued function defined on a domain $D \subset \mathbb{C}$ and has continuous derivatives up to order 2, where \mathbb{C} is the complex plane. We say f is a *harmonic function* if for any $z = (x, y) \in D$, we have $\Delta f(z) = \nabla \cdot \nabla f(z) = \frac{\partial^2 f(z)}{\partial x^2} + \frac{\partial^2 f(z)}{\partial y^2} = 0$, where $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$. Consider a complex function $F = (f^u, f^v)$ composed of two real valued functions, $F : \mathbb{C} \rightarrow \mathbb{C}$, $(x, y) \rightarrow (u, v)$. We say f^u and f^v are *conjugate* if the complex function $f^u + if^v$ satisfies the Cauchy-Riemann equation, namely, $\frac{\partial f^u}{\partial x} = \frac{\partial f^v}{\partial y}$, and $\frac{\partial f^u}{\partial y} = -\frac{\partial f^v}{\partial x}$. A holomorphic function F has its two components f^u and f^v conjugate and harmonic. F is *biholomorphic*, or a *conformal mapping*, if F is bijective and F^{-1} is also holomorphic.

Harmonic and Conformal Maps of Surface. Consider two general Riemann surfaces M_0 and M_1 , solving mapping $\phi : M_0 \rightarrow M_1$ reduces to solving scalar fields defined on M_0 . The generalized Laplace's equation on surfaces is:

$$\Delta_S f = 0, \tag{3}$$

where $\Delta_S f = \text{div}(\text{grad}f)$, namely, the divergence of the gradient of a scalar field f on the surface, is called the Laplace-Beltrami operator. A scalar field with vanishing Δ_S is called a harmonic scalar field. If we define the Dirichlet energy $E_D(\phi) = \frac{1}{2} \int_S \|\text{grad}\phi\|^2$ for each bijective map $\phi : M_0 \rightarrow M_1$, then the minimizers of E_D , whose each component satisfies eqn (3), are called *harmonic maps* from M_0 to M_1 .

Furthermore, consider $\forall p \in M_0$, and $\tilde{p} = \phi(p) \in M_1$, and for any local parameter chart (U, ψ) covering p and local chart $(\tilde{U}, \tilde{\psi})$ of M_1 covering \tilde{p} (namely, ψ and $\tilde{\psi}$ are

local parameterization from patches of M_0 and M_1 to 2D coordinate disks U and \tilde{U} , respectively), if the transformation between the two coordinate systems, $\tilde{\psi} \circ \phi \circ \psi^{-1}$, is holomorphic, then ϕ is a conformal map. Under this mapping, the first fundamental form ds^2 of M_0 scales isotropically, and does not introduce any angle distortion. Thus, a conformal mapping is always harmonic.

In mathematics, the existence and bijectivity of harmonic maps are guaranteed on convex regions, given by Radó's theorem [Schoen and Yau 1997]; the existence of bijective conformal maps on surfaces is guaranteed by the uniformization theorem. They can be computed effectively on discrete triangle meshes [Gu and Yau 2007]. Variations of the approximation to harmonic energies are optimized using discrete Laplace-Beltrami operators in order to construct such maps. Most linear methods work well on a convex Dirichlet boundary condition, where both u and v are solutions to linear elliptic partial differential equations. The harmonic mapping is physically natural, since it minimizes the stretching in the sense that it reduces the Dirichlet energy as much as possible over the surface S . Therefore, it is often used as an effective map between surfaces. However, a harmonic map is not necessarily conformal. And also, it is "one-sided," meaning that the inverse of a harmonic map may not be harmonic, while the inverse of a conformal mapping is still conformal.

Equiareal Mapping. In many medical image analysis problems, since many soft tissues deform nonrigidly with area- and volume-preserving behaviors, people compute equiareal or volume-preserving maps between data to model their change. While conformal mapping preserves the shape and allows only isotropic scaling locally, equiareal mapping, in contrast, allows rotation and shear locally, but prevents scaling transformations. Equiareal maps usually have infinite degrees of freedom. For example, the conformal mapping from a topological disk to a planar disk is unique up to 3 degrees of freedom (thus 1.5 markers uniquely determines a conformal map) between topological sphere surfaces, conformal mapping has 6 degrees of freedom (thus 3 markers uniquely determines a conformal map) [Gu and Yau 2003]. In contrast, equiareal mappings between two given surfaces cannot be determined using a finite number of markers. An infinite amount of equiareal mappings can be obtained under a given boundary mapping condition, and a solution may exhibit extremely large shear and may be highly undesirable. Recently, the technique of Optimal Mass Transport has been used as an effective approach in solving equiareal mapping while maximally preserving angle and local shape of the surface [Dominitz and Tannenbaum 2010; Zhao et al. 2013].

Isometric or Metric-Preserving Mapping. If a map is conformal and also area-preserving, then it is length-preserving or isometric. Geodesic distances on the manifold will not change after the transformation inferred by isometric mapping. Considering the local Jacobian matrix J of an isometric mapping, the two nonzero singular values σ_1, σ_2 of J should both be 1 everywhere. Hence, Hormann and Greiner [2000] measure the isometric distortion using an objective function: $E = \sum_{f \in \Sigma} (\frac{\sigma_{f,1}}{\sigma_{f,2}} + \frac{\sigma_{f,2}}{\sigma_{f,1}})$, where each f is a triangle facet (2-simplex) upon which the piecewise linear mapping function's Jacobian has the singular values $\sigma_{f,1}$ and $\sigma_{f,2}$. In Degener et al. [2003], the isometric distortion is measured using a combined angle and area distortions as $E = E_{angle} \cdot E_{area}^n$, where E_{angle} and E_{area} are the angle and area distortions measured on each triangle, respectively, and n is a weighting factor to balance these two terms.

Quasiconformal Mapping. Quasiconformal map is another emerging technique that allows extra geometric or metric (or area-preserving) constraints. The exact conformal maps between high-genus surfaces rarely exist, unless the two surfaces share the same conformal structure. Therefore, one considers the natural extension of the space of conformal maps to the richer space of quasiconformal maps. Quasiconformal maps with bounded angle distortion and guaranteed local bijectivity can be computed [Weber

et al. 2012; Lui et al. 2012] for solutions of problems with more complicated boundary constraints.

Biharmonic Mapping. Another approach to support the handling of higher-order boundary constraints in mapping design is to solve higher-order partial differential equations in mapping computation. For example, instead of using harmonic scalar fields, biharmonic scalar fields can be used to compose the maps. With biharmonic functions, the continuity of the map on the boundary can be better controlled. Xu et al. [2013a] solve biharmonic volumetric maps using fundamental solution methods, and use them to enforce smooth transitions of the mapping functions along the partitioning boundary of the adjacent subregions.

2.3.3. Classification of Mapping Models Using Surface Topology. To solve the bijective and low-distortion map, different mapping strategies are often adopted when the given shapes have different topologies. Mapping shapes with complicated topology is often much more difficult, and need to be converted to the mapping of shapes with simplified/trivial topology. We classify mapping computation strategies into the following three types:

- Topological-Disk Surface Mapping:* to map surface patches that are topological disks (genus-0, open).
- Spherical Surface Mapping:* to map surfaces that are topological spheres (genus-0, closed).
- High-Genus Surface Mapping:* to map surfaces that have complicated topology (genus- g , $g \geq 1$).

Mapping Topological Disk Surfaces. When given objects are topological disks (have no handles and are homeomorphic to a planar region), they both can be parameterized onto a common 2D planar region $\Omega \subset \mathbb{R}^2$ for shape mapping. The basic idea is to construct $f_1 : M_1 \rightarrow \Omega$ and $f_2 : M_2 \rightarrow \Omega$, then compose the map using $f : f_2^{-1} \circ f_1$ on Ω . The flattening of topological disks has been extensively studied in the parameterization literature [Floater and Hormann 2005; Sheffer et al. 2006; Hormann et al. 2007]. To avoid unnecessary overlaps, we skip extensive discussions but briefly recap a classic pipeline for disc parameterization here and illustrate a comparison of a few representative approaches in Table II.

Mapping with Fixed Boundary. One typical flattening strategy maps topological disks onto a planar region with a predetermined boundary shape. The parameterization is formulated into a problem with two stages: (1) boundary parameterization and (2) interior parameterization. Given the input triangle mesh M and 2D domain Ω , first, compute the mapping from the boundary curve loop ∂M to the domain boundary $\partial \Omega$. Then, solve interior mapping by minimizing some discretized quadratic smoothness energy. Its minimization reduces to solving a sparse linear system, which intuitively enforces the vertex to be mapped to the weighted average of the images of its surrounding vertices: $u_i = \sum_{j \in N_i} w_{ij} u_j$, where N_i denotes the set of one-ring neighboring vertices of vertex- i .

Tutte [1963] shows that if the target domain is convex, and all the weights ω_{ij} are positive, then the obtained parameterization is bijective. He uses the constant unit weights $w_{ij} = 1$ to construct valid flattening for topological disk surface patches. Later, different weights have been developed to minimize different distortions; many of them can be considered as minimizing various potential energies of a string system. For example, the well known cotangent weight [Eck et al. 1995; Pinkall and Polthier 1993] is a discretization of Dirichlet energy using the finite element method, and results in a discrete harmonic map. The weight is $\omega_{ij} = (\cot \alpha_{ij} + \cot \beta_{ij})/2$, where α_{ij} and β_{ij} , as shown in Figure 2, are the opposite angles in the two faces sharing the

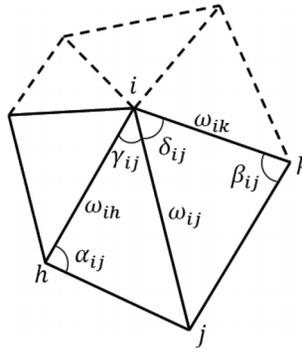


Fig. 2. Notations of angles and weights.

edge (i, j) . However, if the mesh contains skinny triangles (hence, big obtuse angles exist), some ω_{ij} could be negative, and the parameterization can be nonbijective in this local region. The shape preserving parameterization [Floater 1997] and its variance [Guskov 2004], and the mean-value weights [Floater 2003] design the positive weights in the meantime of approximating vanishing Laplacian. The mean-value coordinate [Floater 2003] approximates harmonic functions using their mean-value property, with guaranteed positive weights and bijective functions. The proposed weight is $\omega_{ij} = (\tan(\gamma_{ij}/2) + \tan(\delta_{ij}/2))/\|\mathbf{x}_j - \mathbf{x}_i\|$, where \mathbf{x}_i and \mathbf{x}_j are vertex- i and vertex- j 's the original 3D positions, and γ_{ij} and δ_{ij} are the angles in the two triangles sharing the edge (i, j) , as shown in Figure 2.

Mapping with fixed boundary condition can usually be computed efficiently since solving only sparse linear systems is required. However, the mapping quality relies on the domain boundary shape. When surfaces have complicated geometries, a pre-determined boundary shape may result in parameterizations with large distortion. The *virtual boundary* strategy has been developed [Lee et al. 2002; Zhang et al. 2005] to partly remedy this problem. The virtual boundary is one or several layers away from the real surface boundary so that vertices on the original boundary have better flexibility to move inside this virtual “buffer” zone and reduce the distortion.

Free boundary mapping. Flattening the surface to a domain without predetermining its boundary shape may better reduce the metric distortion near the boundary. Lévy et al. [2002] and Desbrun et al. [2002] derive the free-boundary linear parameterization scheme based on different formulations of conformal energy and end up minimizing the same energy. The Most Isometric Parameterizations (MIPS) method [Hormann and Greiner 2000] optimizes a nonlinear function that measures the mesh conformality and area distortion. It starts from a fixed-boundary harmonic parameterization and iteratively moves the vertex locally to reduce metric distortion. Free-boundary parameterizations often result in less distorted mapping at the cost of using higher-order nonlinear objective functions. In addition, local bijectivity is usually enforced in various free-boundary mapping algorithms, but without enforcing the geometry of the flattened-boundary contours, many such algorithms fail to prevent global self-intersection.

Mapping Surfaces with the Sphere Topology. The boundary surfaces of all solid objects with no handles or voids are *closed genus-0 surfaces*, which are homeomorphic to a sphere. The sphere domain naturally produces seamless parameterizations [Alexa 1999; Asirvatham et al. 2005; Kobbelt et al. 1999] for closed genus-0 surfaces. A spherical parameterization $f : M \rightarrow S^2$ can serve as a key enabling tool to compose mapping between these surfaces.

In Haker et al. [2000], the unit sphere S^2 , after removing one triangle bounding the north pole, is projected to the 2D complex plane using stereographic projection. The surface M , with one triangle removed, becomes a topological disk and is flattened. The two planar embeddings are merged to compose $f : M \rightarrow S^2$. Kobbelt et al. [1999] and Alexa [1999] extend the planar discrete harmonic map to spheres, and directly use Gauss-Seidel iterations to update the mapping of each vertex on the tangent plane on the sphere before normalizing $f(v_i)$ to a unit vector. Isenburg et al. [2001] partition the surface into two parts, then map each of them onto a hemisphere. The partitioning boundary is mapped onto the equator and the interior region is mapped using a barycentric weight [Tutte 1963]. Gu and Yau [2003] solve the conformal spherical harmonic mapping based on the fact that the spherical harmonic map is also conformal, where the 6 degrees of freedom of the Möbius transformation ambiguity can be eliminated by restricting the alignment of 3 feature points. Gotsman et al. [2003] formulate the spherical barycentric formulation based on spectral graph theory; solving a spherical map reduces to solving a quadratic system of equations, which is relatively expensive. [Saba et al. 2005] improve the algorithm numerically so that this system can be solved with better efficiency and stability. Friedel et al. [2005] derive the spherical Dirichlet energy and use it to compute spherical mapping balancing the area and angle distortion. Zayer et al. [2006] compute the spherical mapping using curvilinear coordinates. First, they slice the surface using a “date line” and obtain an initial map that distorts both angles and areas by solving the Laplace equations of the polar coordinates; they then optimize the spherical map using a modified mean-value coordinate algorithm that incorporates both stretching and area distortion. Finally, a local smoothing is performed along the cutting boundary to remove the artifact. Stephenson [2005] uses a circle packing method to construct conformal brain mapping. Jin et al. [2008] use discrete surface Ricci flow to compute spherical metric and embedding for genus-0 closed surfaces. Sheffer et al. [2004] developed a spherical *angle-based-flattening* (ABF) algorithm that first solves a valid metric (edge lengths) on a sphere using the spherical cosine and sine laws, then embeds the mesh on the sphere. Lee et al. [2006] map genus- g surfaces to a set of $g + 1$ spherical domains, where each handle glued through its tunnel loop is mapped to a sphere: the base domain is called a positive surface while the remaining g handle patches are referred to as negative surfaces. Yu et al. [2010] parameterize general genus- g surfaces M onto the unit sphere by first slicing all the topological handles through their handle or tunnel loops [Dey et al. 2007], then mapping the resultant genus-0 surface with g filled boundaries onto a sphere. With such a topological operation, M is parameterized onto a unit S^2 with g small regions removed. Zeng et al. [2007] solve the spherical parameterization of high-genus surfaces using the meromorphic function defined on the surface. They conformally wrap a genus- g surface onto a sphere by several layers through $2g + 2$ branch points, which are singularities shared by two adjacent layers.

Multi-resolution techniques have been used in spherical parameterization to improve its efficiency. Shapiro and Tal [1998] embed meshes onto a spherical domain with pure topological operations without distortion minimization, and Praun and Hoppe [2003] start from a multiresolution decomposition, then alternatively add each vertex and refine their positions inside their neighborhood. The stretch metric is minimized in this process. Asirvatham et al. [2005] also solve a constrained spherical parameterization using the progressive mesh; feature point alignment is enforced on the coarsest resolution on the sphere. Wan et al. [2012, 2013] further develop an improved hierarchical optimization scheme using progressive meshes to optimize the metric, combining the energy of Asirvathan et al. [2005] and Praun and Hoppe [2003]; their adaptive node selection and great circle searching approach result in a more efficient and lower-distorted mapping computation.

Mapping High-Genus Surfaces. For high-genus surfaces, finding a simple canonical domain such as spheres or planar disks becomes nontrivial. There are two general approaches to computing their cross-surface parameterization.

The local approach is to first segment surfaces M_i , $i = 1, 2$ into two consistent sets of subregions, $M_i = \bigcup_{j=1}^m \tilde{M}_{i,j}$, then obtain global mapping by composing local maps between corresponding subregions: $f_k : \tilde{M}_{1,k} \rightarrow \tilde{M}_{2,k}$. To enable such a divide-and-conquer strategy, the decomposition should satisfy these conditions: that (1) each subregion has simple topology (e.g., topological disks) and geometry (e.g., flat, convex, and so on), so that the subregion mapping can be computed efficiently; and (2) the partitioning is topologically consistent, that is, the dual graphs of the two decompositions are isomorphic to each other. This decomposition is referred to as *consistent decomposition* (see Section 3.1.2).

Earlier divide-and-conquer techniques perform consistent decomposition manually. In the algorithm of DeCarlo and Gallier [1996], users specify a base mesh manually. With consistently designed base meshes between given surfaces, cross-surface mapping can be composed directly. However, this approach requires tedious and careful user specification. In addition, when the input surfaces have complicated topology or geometry, the design can become nonintuitive and difficult. Gregory et al. [1998] and Zöckler et al. [2000] also construct cross-shape mapping based on manually constructed base meshes. After the common base mesh is labeled by the user, corresponding subregions are matched via harmonic maps or barycentric maps. There are a few more algorithms relying on such manual base mesh construction, mainly with applications in morphing construction, as discussed in the survey article of Lazarus and Verroust [1998].

More recently, a few semi-automatic or automatic consistent decomposition algorithms have been developed [Kraevoy and Sheffer 2004; Schreiner et al. 2004; Li et al. 2008b, 2009; Kwok et al. 2012; Zhang and Li 2012; Shalom et al. 2008; Kraevoy et al. 2007; Zhang et al. 2014; He et al. 2009a]. Kraevoy and Sheffer [2004] and Kwok et al. [2012] construct the consistent partitioning of multiple surfaces following user-specified feature points. Their limitations are that a large amount of features need to be carefully specified for robust tracing. Praun et al. [2001] developed an algorithm to trace consistent partitioning boundaries on different triangle meshes; it can transfer an existing mesh partitioning to another topologically equivalent surface. Schreiner et al. [2004] first trace consistent sets of triangular patches on multiple surfaces using progressive mesh representations, having feature points as endpoints of these tracing paths; then, starting with the map of the coarse base mesh, it iteratively refines the map to higher resolutions together with the vertex split. These algorithms are semi-automatic as consistently specified features are necessary input from the user. Li et al. [2008b] and Zhang and Li [2012] develop automatic consistent surface decomposition frameworks by using the so-called canonical *pants decomposition*. Such a pants decomposition scheme can partition surfaces with nontrivial topology into topologically simple patches (also with a simple adjacency relationship). Furthermore, it can flexibly compute the consistent decomposition of two surfaces that have different topology. Bennett et al. [2008] also introduce a robust topological approach to construct maps between two surfaces (triangle meshes) with possibly different genres.

Mapping computation through divide-and-conquer is efficient, and locally the mapping quality can be good. However, distortion and discontinuity across the partitioning boundary could be significantly larger than the interior regions; certain smoothing or relaxation operations are needed to remove these artifacts. In addition, the effective partitioning for shapes of arbitrary topology/geometry is very challenging, and the mapping result is greatly affected by the quality of this problem decomposition.

The global approach, in contrast, is to directly compute global mapping through one global domain with certain desirable Riemannian metrics. For example, the domain Ω for high-genus surfaces can be Riemannian surfaces of nonpositive constant curvature [Schoen and Yau 1997] by deforming the target surface \mathcal{N}_2 to tile the complex plane \mathcal{C}^2 (if surface \mathcal{N}_2 is genus-1) or tile the unit hyperbolic disk \mathcal{H}^2 (if \mathcal{N}_2 is genus- n , $n > 1$). Upon such a Ω , a harmonic map can be computed through a global optimization. A key advantage of using such a uniformization metric is that the objective function does not have local minima and one can obtain a globally optimal solution [Li et al. 2008a]. Other types of canonical domain Ω with good geometric regularity/simplicity, such as polycubes [Wan et al. 2011; Yu et al. 2013; Yu and Li 2014] or N-hole tori [Grimm and Hughes 2003], can also be used to compose cross-surface mapping. Note that, again, the global mapping of a high-genus surface onto a canonical domain can be obtained through both direct and indirect approaches. The direct algorithms [Boier-Martin et al. 2004; Dong et al. 2005] solve the mapping directly by finding the scalar values u, v defined on surfaces. The indirect algorithms first solve the derivatives [Gu and Yau 2003; Gortler et al. 2006; Tong et al. 2006; Ray et al. 2006] of the map or solve the flat Riemannian metric [Collins and Stephenson 2003; Kharevych et al. 2006; Jin et al. 2008] before obtaining the map through integration or embedding. Another global mapping computation approach is to solve an energy-minimized diffeomorphism in a metric shape space. The idea is to model the mapping of one shape from the other via a dynamic flow of diffeomorphisms (namely, time-dependent deformation) $t \in [0, 1]$ of the embedded ambient space \mathcal{R}^d . The Large Deformation Diffeomorphic Metric Mapping (LDDMM) [Beg et al. 2005] is this kind of approach, which solves not only a bijective map between shapes but also define a metric distance in this constructed space.

For geometric shapes with nontrivial topologies, many recent parameterization algorithms can flatten them onto canonical parametric domains [Gu and Yau 2003; Ray et al. 2006; Bommes et al. 2013], however, they inevitably introduce singularity points that are determined by both surface topology and geometry. Due to the inconsistent existence of these singularities, many cannot be easily adopted directly to compose a bijective intersurface map between given high-genus shapes.

Global approaches solve a smooth mapping between shapes. However, global approaches are usually sensitive to the topology of the data, less flexible in incorporating semantic landmarks, and often computationally expensive.

A list of shape parameterization algorithms that map surfaces onto canonical geometric domains is given in Table II. In the column of Robustness, we discuss the algorithm's robustness against *Mesh* quality, *Topological* noise, and *Geometric* perturbations. The notation, for example, $Y_M N_T N_G$ means that the algorithms are robust against mesh quality, but sensitive to topological noise and geometric perturbation. Note that most fixed-boundary mapping techniques are sensitive to the boundary geometry missing, thus they are considered to be N_G .

2.3.4. Mapping Volumetric Shapes. Many real-world data are volumetric and have both boundary surface geometry and interior texture or material attributes. To model such volumetric data, their interior material, intensity, or other structural information should be considered, and volumetric parameterization needs to be studied. Inter-volume mapping can be extended from the boundary surface mapping. Given two solid regions M_1 and M_2 , the volumetric mapping $f : M_1 \rightarrow M_2$ consists of three scalar fields (ϕ_1, ϕ_2, ϕ_3) defined on M_1 . To measure the *quality* of Φ , we consider two criteria: (1) *Geometric distortion*. Similar to the surface map, it is also desirable to reduce metric distortion and to have the geodesic distances, angles, and volumes preserved under the mapping. (2) *Feature alignment*. The volumetric data usually possess nonuniform interior materials/layers, and their mapping should incorporate these nonuniform

Table II. Classification of Algorithms that Map Surfaces onto Canonical Domains

Method	Distortion	Generality	Boundary	B	C	R
Graph embedding [Tutte 1963]	None	Disk	Fixed	Y	L	$Y_M N_T N_G$
Discrete harmonic map [Eck et al. 1995]	Angle	Disk	Fixed	N	L	$N_M N_T N_G$
Shape preserving [Floater 1997]	Angle	Disk	Fixed	Y	L	$Y_M N_T N_G$
Mean-value [Floater 2003]	Angle	Disk	Fixed	Y	L	$Y_M N_T N_G$
LSCM/DCP [Lévy et al. 2002; Desbrun et al. 2002]	Angle	Disk	Free	N	L	$N_M Y_T Y_G$
MIPS [Hormann and Greiner 2000]	Angle	Disk	Free	G	N	$Y_M Y_T Y_G$
ABF [De Sturler and Sturler 2000; De Sheffer et al. 2005]	Angle	Disk, sphere	Free	L	N	$N_M Y_T Y_G$
Circle packing [Bowers and Hurdal 2003]	Angle	Disk	Free	L	N	$N_M N_T Y_G$
Stretch minimizing [Sander et al. 2001]	Distance	Disk	Free	G	N	$N_M N_T Y_G$
Stereographic projection [Haker et al. 2000]	Angle	Sphere	Fixed	N	L	$N_M N_T Y_G$
Spherical discrete harmonic map [Gu and Yau 2003]	Angle	Sphere	Free	N	N	$N_M N_T Y_G$
Spherical barycentric map [Gotsman et al. 2003; Saba et al. 2005]	Angle	Sphere	Free	N	N	$Y_M N_T Y_G$
Progressive mapping [Praun and Hoppe 2003; Wan et al. 2012]	Stretch	Sphere	Free	N	N	$Y_M N_T Y_G$
Curvilinear coordinates [Zayer et al. 2006]	Angle	Sphere	Fixed	N	N	$N_M N_T Y_G$
Optimal mass transport [Dominitz and Tannenbaum 2010; Zhao et al. 2013]	Area	Disk, sphere	Free	G	N	$Y_M N_T Y_G$
Progressive mapping [Khodakovskiy et al. 2003]	Angle	General	Free	Y	N	$Y_M N_T Y_G$

Each column indicates the categories they belong to: **Distortion** to minimize, **Generality**, **Boundary** shape, **Bijectivity** (besides the absolute **Yes**, and **No**, some methods only guarantee **Local** bijectivity and others have only **Global** bijectivity), **Complexity** (**Linear** or **Nonlinear**), **Robustness** against mesh quality, topological noise, and geometric perturbation.

structures. For example, feature curves, feature surfaces, and other substructures (such as local landmarks or global symmetry pattern) often encode important information. Therefore, the scalar fields $\{\phi_j, j = 1, 2, 3\}$ may need to satisfy some extra constraints, and their gradients $\{\nabla\phi_j\}$ should align with (be parallel or perpendicular to) some given directions.

Wang et al. [2004] discretize the volumetric harmonic energy over tetrahedral mesh, and generalize the surface spherical parameterization to the harmonic volumetric spherical parameterization. Such a discrete Laplacian discretization was also used for extending polycube surface parameterization to the mapping of a given model's entire volume space [Han et al. 2010; Xia et al. 2010]. Li et al. [2007, 2010] and Xu et al. [2013a] use surface mapping as the boundary condition and extend the cross-surface map to a cross-volume map using the method of fundamental solutions. Martin et al. [2008] parameterize the surface model onto a cylinder, then extend the parameterization into the interior volume using the finite element method (FEM); then, they generalize this FEM solver to more complicated models that can be approximated as shapes with medial surfaces [Martin et al. 2012]. There are also quite a few volumetric parameterization algorithms that directly solve the mapping between two volume regions without first computing boundary surface mapping. Such an approach, avoiding precomputing a boundary condition, can usually result in smaller mapping distortion.

However, the problem reduces to a big nonlinear optimization problem with both integer and certain linear/nonlinear constraints, thus is much more costly to solve [Nieser et al. 2011; Huang et al. 2011; Li et al. 2012]. More importantly, these mapping algorithms [Nieser et al. 2011; Li et al. 2012] allow the insertion of singularities during the mapping computation. But freely distributed singularities (that are not consistent) are usually not permitted in a valid inter-shape correspondence, therefore, these parameterizations are usually used in volumetric remeshing and not directly applicable for cross-shape mapping.

2.4. Completeness of Mapping Domain

Cross-shape mapping can be classified by the completeness of the mapping domains. The *complete mapping* between the given shapes M_1 and M_2 seeks for a bijective map $f : M_1 \rightarrow M_2$ that corresponds M_1 and M_2 completely, as defined earlier; while the *partial mapping* between M_1 and M_2 , finds a common region under the map, that is, $f : N_1 \rightarrow N_2, N_1 \subset M_1, N_2 \subset M_2$. Partial mapping is useful in pattern/template detection, fusing/stitching partially created/obtained data [Li et al. 2008], and so on. Due to the freedom allowed in partial mapping, one usually needs to restrict the transformation with certain rigidness/smoothness [Li et al. 2008, 2009; Hou and Qin 2010] or utilize feature correspondence constraints [Allen et al. 2003] to control the map. More about feature modeling and correspondence will be discussed in Section 3.

3. CONSTRAINTS: FEATURE MODELING AND ALIGNMENT

To map two given geometric regions M_1 and M_2 with the same topology, one can choose suitable shape representations and directly solve a mapping with minimized metric distortion/geometric deviation/smoothness energy. In some scenarios, the given shapes are incomplete (so that a partial mapping is needed) or geometrically complex (so that a global mapping is difficult or costly to solve); effective mapping computation algorithms often use the guidance of *shape features*. Shape features can be generally defined as certain salient structures, points, or subregions of the shapes that are prominent according to a particular definition of interestingness or saliency. They encode important geometric invariants or properties of shapes for efficient matching computation. A common pipeline for feature-guided mapping computation first extracts these features on both models, then establishes correspondence between features of M_1 and M_2 , and finally enforces such feature correspondence in mapping computation.

Effective geometric features, incorporating various global and local invariants, structures, and saliency, can facilitate shape mapping. We generally classify 3D shape features according to the scale they encode the shapes as summarized in Table III.

- Global Structure or Quantities*. Global topological and geometric invariants are important shape indicators. Basic geometric properties, such as mass center, shape volume, and principal axes, are usually extracted and used to normalize shapes for their mapping. Other global properties, such as moments, topologies, media axes, and segmented subparts, are also effective global features to incorporate in mapping computation.
- Local Features*. Local geometric properties encode local shape saliency. Widely used local features include semantic landmarks, geometric derivatives such as curvature, convexity/concaveness, crest lines, and tips/extremal points. With local feature correspondence, mapping can be computed following a coarse-to-dense matching refinement scheme: first extract and match local features to roughly correlate the shapes, then extend such coarse correspondence to pointwise dense matching.

Table III. Feature-Based Shape Representation

Global-Scale Features	Global Invariants	Section 3.1
	Medial-Axis Representation	Section 3.1.1
	Segmentation & Intersubpart Relationship	Section 3.1.2
Local-scale features	Local geometricsaliency	Section 3.2
Multi-scale features	Features in scale-space	Section 3.3

—*Multiscale or multiresolutional features.* Features can be defined in a multiscale manner to encode both local and global properties. Various spectrum-based and wavelet-based shape representations are defined on multiscale features.

3.1. Global Geometric Invariants

Many basic geometric invariants can be used as effective parameters of 3D shapes to correlate shapes undergoing specific deformations. These global geometric invariants can be utilized with other shape properties to facilitate the shape matching.

—*Topological Invariants.* Topological invariants are properties of a shape that is invariant under homeomorphism, or continuous deformations. One can use the Betti number [Hatcher 2002] to encode the topology of the shape. For a nonnegative integer k , the k th Betti number $b_k(M)$ of the manifold M is the rank of the k th homology group of M . Intuitively, b_0 indicates the number of connected components; b_1 indicates the number of handles, and b_2 indicates the number of voids. For example, for a connected surface M , the first Betti number $b_1(M)$ is also the well known Euler Characteristic number χ . If M is represented using a simplicial complex (triangle mesh), $\chi = F - E + V$ where F , E , V are numbers of faces, edges, and vertices, and it encodes the topology of this surface $\chi = 2 - 2g - b$ where g is the genus (i.e., the number of topological handles) and b is the number of boundary loops. These topological indices encode the lowest-level fundamental properties of shapes, but they are sensitive to topological noise such as boundaries and topological shortcuts.

—*Geometric Invariants.* A few geometric invariants are basic global properties of 3D shapes. These invariants, rather than being adopted directly as complete shape descriptor, are often integrated into other shape features for shape matching or used to normalize or pre-align 3D shapes before mapping computation. These global shape properties are usually not sensitive to local noise and geometric perturbations.

—*Mass Center.* The mass center of a shape is also called the centroid or center of gravity. For uniformly sampled point clouds, it can be simply computed by the average of all the point coordinates; for non-uniformly sampled points, the density or area/volumes of regions surrounding points should be incorporated as weights in the computation. The mass center is generally invariant under rotation, and robust against sampling and noise. To make the shape matching algorithm translation-invariant, we may pre-normalize the given shape by moving its mass center to the origin.

—*Principal Axes, Axes of Least Inertia.* The *axis of least inertia* (ALI) is a reference line describing the orientation of the geometry. It can be defined as an optimal line from where the integral of the square of the distances to all the points on the shape boundary is minimized. Since the ALI passes through the centroid, one can normalize the shape [Horn 1987] by rotating ALI to a coordinate axis. Shapes can be rigidly prealigned using principal component analysis (PCA): the three principal axes of the centroid-normalized shape are the eigenvectors of the covariance matrix composed of point coordinates.

—*Bounding Volume.* The 3D shape can also be approximated using a bounding volume with relatively simple geometry. From simple to complex, effective bounding

volumes include bounding spheres, bounding ellipsoids, bounding boxes, discrete oriented polytopes (DOPs) [Held et al. 1996], and convex hulls. Besides the bounding volume itself, the integrated geometric variance from the shape to its bounding volume is also a global shape descriptor.

- Geometric shape factor*: Various geometric aspect ratios of a shape’s bounding volumes can serve as global geometric invariants. For example, the *eccentricity* encodes the ratios of the lengths of principal axes; the *circularity ratio* is the ratio of volume (or boundary area) of the shape to the volume (or boundary area) of its bounding sphere; the *cubeoid ratio* is the ratio of volume (or boundary area) of the shape to the volume (or boundary area) of its bounding box. These ratios characterize the degree of variation of the shape from its fitted bounding volume.
- Shape Spectra*. The shape’s spectrum can be used as a global shape feature. The spectra can be calculated on any representation of the geometric shapes from parameterized representations such as NURBS/meshes to implicit functions. Reuter et al. [2005] used the Laplace-spectra for 3D shape matching. The spectrum is often intrinsic (location independent) and isometry-invariant. The *Global Point Signature* (GPS) [Rustamov 2007; Ovsjanikov et al. 2008] is another spectral descriptor that captures the global geometric properties. It uses the eigenvalues of the Laplace-Beltrami operator together with the corresponding eigenfunctions to encode the geometric shapes. Spherical harmonics provide a rotation-invariant representation [Kazhdan et al. 2003] for solid shapes. With spherical parameterization, the coefficients of the spherical harmonic basis functions form an intrinsic shape descriptor for genus-0 surfaces [Huang et al. 2005; Wan et al. 2012]. The *manifold harmonics* [Vallet and Lévy 2008] is discrete generalization of the Fourier transform on a surface mesh. Likewise, 3D Zernike descriptor [Novotni and Klein 2003] converts the shapes to a multiresolutional representation using the functional spaces for efficient shape analysis and matching. Jain and Zhang [2006] construct an affinity matrix of a shape using its geodesics, then find the spectral embedding of the matrix so that the 3D mesh is transformed from the spatial domain to the spectral domain for matching purposes. Spectrum-based shape representation is often sensitive to geometric and topological noise.
- Geodesic Distances*. The geodesic distances measure the length of the shortest path along each pair of points on the surface; as fundamental properties related to the surface metric, they can be used as an effective global feature in shape analysis and matching [Hilaga et al. 2001; Schmidt et al. 2006; Bronstein et al. 2007; Tevs et al. 2009; Tung and Matsuyama 2010; Ying et al. 2013]. However, geodesics are usually sensitive to topological noise: a small topological shortcut may result in a significantly large change of geodesic distance and path. Most of the existing discrete geodesic algorithms can be applied only to noise-free meshes. Geodesic computation algorithms [Quynh et al. 2012] are also developed to compute meaningful approximate geodesics on polygonal meshes with holes without explicit hole filling.
- Other Intrinsic Distances*. A few other intrinsic *diffusion distances* based on diffusion geometry, such as heat diffusion [Sun et al. 2009; Bronstein and Kokkinos 2010], diffusion distance [Coifman and Lafon 2006], random-walk distance [Fouss et al. 2007], and biharmonic distance [Lipman et al. 2010], have also been proposed. Diffusion geometry appears to be more robust to topological noise [Bronstein et al. 2010], which does not change significantly with the addition of geometric noise or topological changes. Also, it allows definition of intrinsic shape similarity that is invariant to inelastic deformations, which is desirable for nonrigid shape registration between different models.

Besides these global invariants and features, two other widely used global geometric features are *medial axes* and *intersubpart structures from shape decomposition*;

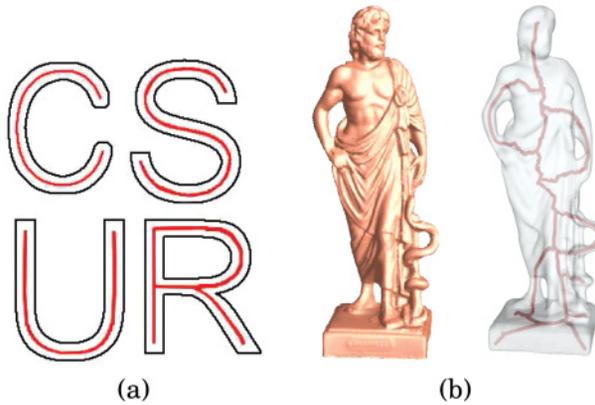


Fig. 3. Skeletons of a few 2D and 3D surfaces. The Greek sculpture model in (b) is provided courtesy of IMATI by the AIMSHAPE-VISIONAIR Shape Repository.

both can often effectively facilitate mapping computation. We elaborate on them in Sections 3.1.1 and 3.1.2.

3.1.1. Geometric Structure from Medial Axis. The *medial axis* [Blum 1967], sometimes intuitively referred to as the *shape skeleton*, is a 1D graph representation for 2D or 3D models (Figure 3). It is an effective global feature to encode the shape’s topological and geometric characteristics and to guide shape mapping.

A *medial axis* of an n -D shape is the locus of points that have two or more closest points on the shape boundary (e.g., boundary contour for a 2D region or boundary surface for a 3D solid object). In 3D, such a locus may become a surface, which is then also referred to as a *medial surface* [Gong and Bertrand 1990]. A classic approach for computing the medial axis or medial surface is the *grassfire* algorithm: (1) Plant dry grass all over the 2D/3D region M , which is bounded by contour curves/surfaces ∂M ; (2) Start a grass fire on ∂M at the same time and propagate with a same speed; (3) The loci of points where the fire fronts meet is the medial axis/surface. Such a process conducted in a geometric region in \mathcal{R}^n generally results in a medial axis of dimension $n - 1$, in other words, this grassfire algorithm can result in getting a “surface-like” skeleton of a 3D shape. But a one-dimensional curve feature is a more efficient encoding for shape mapping [Svensson et al. 2002] and is preferred. Variants of the medial axis definition have been used to obtain curve skeletons. For example, Dey and Sun [2006] define the curve skeleton for 3D shapes as a subset of the medial surface using the extreme values of a medial geodesic function on the medial surfaces.

The process to extract the spatial skeleton of a given 3D shape, often referred to as *skeletonization*, should preserve different criteria for different applications. The following are a few criteria that are often considered in shape mapping. More detailed discussions on general criteria of skeletonization can be found in Siddiqi and Pizer [2009] and Cornea et al. [2007].

—*Homotopy.* A skeleton $K(M)$ is homotopic (Figure 3(a)) if M can contiguously deform (shrink) to $K(M)$ without tearing (see its formal definition in Hatcher [2002]).

A homotopic skeleton can correctly encode the topology (specifically, connected components, tunnels, and cavities) of the shape, thus it is desirable to extract homotopic skeleton features in shape mapping. Following this definition, a homotopic skeleton can be extracted from such a continuous shrinking of the given shape.

—*Invariance under isometry or other transformations.* When the given shape M deforms under some basic geometric transformation T , if the skeleton of the deformed object

$T(M)$, denoted as $K(T(M))$, also differs from the original skeleton $K(M)$ by the same transformation, namely, $K(T(M)) = T(K(M))$, then the skeleton is invariant under this transformation T . When tracking spatial and temporal deforming 3D objects, this is necessary in order for the skeleton to preserve the intrinsic structure of the object, regardless of where and how the object embeds in the space.

- Reconstruction*. If we can recover M from the skeleton K , we say K has the capability of reconstruction. For example, by recording radii of maximal inscribed balls on K skeletal nodes, M might be reconstructed from K [Blum 1967]. It characterizes with how much detail the skeleton captures the geometry of the shape.
- Centeredness*. It is important for the skeleton to be in the center of the object. Centeredness is important in applications such as shape compression, vortex core extraction, virtual navigation, and morphing animation. However, perfect centeredness requires the skeleton to completely lie on the medial surface, which makes it sensitive to noise. Therefore, approximate centeredness is often preferred in guiding shape mapping.
- Component-wise differentiation*. Skeletonization is closely related to shape decomposition, and should distinguish different components and reflect the shape's part structure; for example, logical components correspond one-to-one to skeletal nodes or arcs. With this property, the mapped skeletons can infer the mapping of corresponding subparts.
- Robustness against noise*. A skeleton should be robust against noise so that when shapes undergo deformations that cause local geometric variance, its skeletons can remain stable and can guide the mapping computation.

We briefly recap the extraction of skeleton features for shape mapping computation in the following.

- Thinning Approaches*. Following the definition of medial axis, this approach simulates the *grassfire* flow to contract the shape using various advanced principles. Its advantage is that it is efficient and parallelizable, but different propagation schemes may lead to different skeletonization results.
- Distance-Field Approaches*. The distance field of a 3D object M encodes the smallest distance from each interior point to the shape boundary ∂M . Various distances such as Euclidean 2-norm [Malandain and Fernandez-Vidal 1998] or other metrics [Borgefors 1996; Dey and Sun 2006] can be used to construct the distance fields. The “ridges” of each distance field identify points on local maximal ball loci, that is, potential skeletal nodes. Ridge points can be extracted, pruned, then connected to form the final skeleton.
- Potential-Field Approaches*. Instead of using the distance function, other fields can be used for skeletonization. The basic idea is to construct a field where the potential at a point \mathbf{x} is approximated by a weighted sum of potential functions determined by both the shape boundary and \mathbf{x} . The potential kernel functions could be various radial basis functions [Ma et al. 2003] or fundamental solutions [Li et al. 2007]. Similar postprocessing is needed to extract skeletons from these fields.
- Reeb Graph Methods*. Reeb graph methods are topological methods rooted in Morse theory [Milnor 1963]. A Reeb graph is a 1D structure whose vertices are critical points of a real-value, level-set function defined on the model. The connectivity of this graph encodes the topology of the original shape by following the evolution of the level sets of this function. Hilaga et al. [2001] compute a Multiresolutional Reeb Graph (MRG) on surfaces using normalized geodesic distances and suggest a shape-matching scheme through the matching of the graph structure. Tung and Schmitt [2005] improve this method in an *Augmented Reeb Graph* by suggesting a new topological coherence condition to improve graph matching. Pascucci et al.

Table IV. Skeleton Extraction Algorithms and Skeleton Properties (**H**omotopy, **T**ransformation Invariance, **R**econstruction, **C**omponent-wise Differentiation, **R**obustness against Noise, **E**fficiency)

	H	I	R	C	N	E
Thinning	G	F	B	B	F/B	G/F
Distance Field	B	G	B	B	F	G
Potential Field	B	G	B	B	G/F	B
Reeb Graph	G	F	B	B	F/B	G
Decomposition	G	F	F	B	Y/F	F/B

G, F, and B represent whether the algorithms are good, fair, or bad in preserving such properties.

[2007] developed an efficient online algorithm that constructs a Reeb graph for large simplicial meshes using a stream of local updates. The height function, despite its simplicity, often leads to many local minima that complicate the Reeb graph; in contrast, smooth scalars such as harmonic functions [Patane et al. 2008; Wang et al. 2009] can be used to construct a Reeb graph. Due to the maximum principle, the harmonic scalar field has very few critical points and leads to simple Reeb graphs, which could help ease the Reeb graph matching [He et al. 2009b].

—*Shape-Decomposition Methods*. Shape decomposition can extract the intersubparts relationship and whose dual structure infers a skeletal graph. Katz and Tal [2003] decompose the shape and link the components to construct the skeleton; Dey and Zhao [2002] and Attali et al. [2009] extract the skeleton from Voronoi diagrams.

Table IV compares these five common skeletonization algorithms on how well they possess the common properties desirable for shape mapping.

3.1.2. Geometric Structure from Shape Decomposition. The decomposition of a geometric object encodes the spatial relationship among salient subparts of the object. In computer vision, many recognition algorithms identify objects through matching recognizable subparts and their spatial interrelationship.

Suppose M is a k -complex, and S contains k -dimensional subcomplexes of M . A *decomposition* Σ of M is a partition $\Sigma = \{M_1, \dots, M_n\}$, $M_i \in S$, such that $M = \bigcup_{i=1}^n M_i$, $M_i \subset M$, and $M_i \cap M_j = \emptyset$ if $i \neq j$, $\forall i, j = 1, \dots, n$.

The *minima rule* proposed by Hoffman et al. [1984] states that human perception intuitively partitions objects at areas of concavity. Therefore, such a partitioning infers an intrinsic structure of the shape and can sometimes benefit shape-mapping computation. Thorough surveys on shape segmentation techniques are given in Shamir [2008] and Agathos et al. [2007], and data benchmarks [Chen et al. 2009] were built up for evaluating different segmentation methods.

We classify the shape decomposition algorithms into five general categories according to their computation strategies, and briefly discuss their properties in facilitating shape mapping: (1) Many decomposition algorithms adopt the bottom-up scheme, which simultaneously grows preselected subparts while preserving local properties; these methods are also referred to as region growing approaches. (2) In contrast, some other algorithms decompose models through a top-down strategy, which iteratively or hierarchically partitions complex models into subparts. (3) Besides direct domain partitioning methods, shape decomposition can also be obtained from other geometric features such as skeletons or using other implicit methods. (4) More closely related to 3D shape mapping, we also discuss *consistent segmentation*, which partitions multiple models into subparts with coherent adjacency relationships.

Decompositions through Bottom-Up Approaches. A classic approach for partitioning discrete data is through a bottom-up and locally greedy strategy, which starts from several seeds and continuously grows their associate patches/subparts outward until

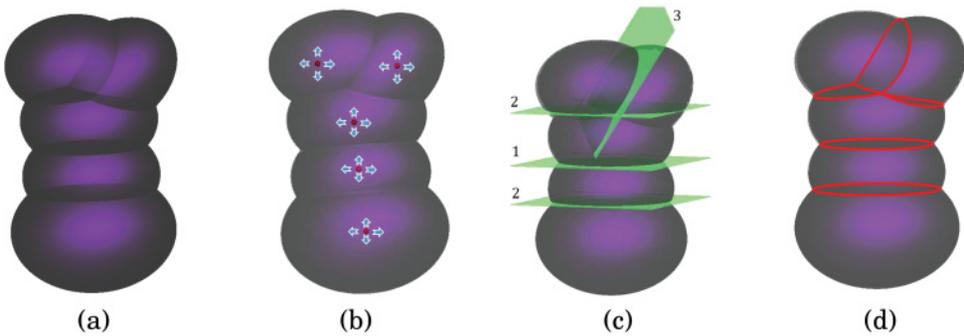


Fig. 4. Segmenting a surface (a) using different strategies: (b) Bottom-up, region-growing (e.g., Lavoue et al. [2005]), (c) Top-down, hierarchical partitioning (e.g., Katz and Tal [2003] and Lien and Amato [2006]), (d) Feature contours tracing (e.g., Lee et al. [2005a]).

all elements of the shape are clustered. This strategy is usually also referred to as *region growing*. Figure 4(b) illustrates this idea on a simple shape (a). Different region-growing algorithms adopt different strategies in (1) seed selection and (2) criteria for determining whether or not neighboring elements should merge into the current cluster (i.e., when to permit growing and when to terminate growing).

Vieira and Shimada [2005] select seeds by noisy levels and Gaussian/mean curvatures of mesh points, and grow subpatches via fitting neighboring regions to bivariate Bézier surfaces. Lavoue et al. [2005] segment surfaces by randomly picking seeds and then clustering constant curvature regions. Zuckerberger et al. [2002] compute the decomposition with a flooding of the dual graph using breadth-first or depth-first search. Faces are consistently collected unless the convexity of a subregion is violated. A potential oversegmentation problem is then tackled by merging smaller subregions to neighboring larger ones. These ideas were generally described in Chazelle et al. [1997] as randomly selecting starting faces and then imposing convexity during the growing, and it is improved in Krayevoy et al. [2007] by starting the region growing process with a set of approximately convex parts. Krayevoy and Sheffer [2006] measure convexity of subregions, and stop the growing when a combined threshold of convexity and compactness is reached; while Yamauchi et al. [2005a] use Gaussian curvature as the criterion. Lai et al. [2009] grow the seeded subregions using random walks, which is extended from Grady [2006], guided by a probability model that indicates the probability of a random walk starting from a point (faces) that hits each seed before reaching other seeds.

Watershed Algorithms. The well-known watershed algorithm, widely used for image segmentation at first and extended to 3D meshes in Mangan and Whitaker [1999], can also be considered as a region-growing approach. The watershed algorithm often determines the seeds using the local minima of a specific height function on the mesh, then grows the regions around seeds (flooded basins) incrementally until they reach ridges or maxima of the function. This flooding can be conducted on various height functions (e.g., the Gaussian, mean, or other curvatures, the average geodesic distance [Zhou and Huang 2004], electrical charge distribution [Wu and Levine 1996], and many others [Mangan and Whitaker 1999]).

Region Clustering. Another region-growing strategy is called region clustering, whose computation is through iterative clustering. Garland et al. [2001] use hierarchical clustering to segment shapes with close-to-plane subregions, judging the planarity, orientation bias, and shape bias measures. Inoue et al. [2001] propose a similar approach that clusters subregions based on area size, boundary smoothness, and region flatness.

Attene et al. [2006] use the similar pipeline to cluster approximated plane, sphere, or cylinder primitives. Shlafman et al. [2002] cluster dihedral angles and geodesic distance of the barycenters using k-means clustering. Yamauchi et al. [2005b] apply mean shift clustering to surface normals, which could oversegment a model. Gelfand and Guibas [2004] use “slippable motions” to guide their clustering, and the algorithm is suitable to extract standard linear or quadric patch templates.

Bottom-up algorithms can produce geometrically desirable local subparts but often lack effective control on the global partitioning structure. The adjacency relationship of an intersubpart is difficult to control or predict during the growing process. Therefore, when applying this algorithm on multiple models, the layouts of computed decomposition on two different objects are usually not consistent. But to use decomposition to partition the mapping problem, a consistent decomposition layout for multiple objects, with preferably simple topological adjacency relationship among neighboring subregions, is usually needed. Therefore, in shape mapping, global control of the partitioning structure is preferable.

Decompositions through Top-Down Approaches. A top-down strategy in shape decomposition may offer better control of the global structure of the decomposition. Starting from one root representation of the whole object, it constructs a tree and recursively partitions it into parts until every part meets a certain tolerance. Figure 4(c) illustrates the idea of top-down partitioning.

Katz and Tal [2003] decompose shapes by a hierarchical fuzzy k-means algorithm, based on criteria similar to Shlafman et al. [2002]; the partition algorithm determines the cutting along the fuzzy areas by a minimum cut algorithm guided by deep concavities, conforming to the minima rule previously mentioned. The *Approximate Convex Decomposition* (ACD) algorithm [Lien and Amato 2006] conducts a top-down approach to decompose a given region to approximated convex domains. ACD measures the convexity as the cutting criterion and splits apart when the convexity does not meet the criterion. With a similar processing pipeline, Lien et al. [2006] integrated the measurement of skeleton quality to simultaneously compute shape decomposition and skeletonization. Benkó and Várady [2002] segment point clouds using sequential top-down recursive tests. Lai et al. [2006] combine integral and statistical quantities derived from local surface geometry and develop a segmentation algorithm that has better robustness on meshes with noise. Podolak et al. [2006] present a hybrid algorithm of iterative clustering and graph cut to extract the segmentation suitable for shape matching and automatic viewpoint selection.

Spectrum Based algorithms. We classify another category of segmentation algorithms that are based on spectral graph analysis as another top-down approach. Spectrum-based segmentation methods also start with analyzing the global structure of a shape by correlating the characteristics of a graph and algebraic properties of the Laplacian of the shape. Liu and Zhang [2004] use a symmetric affinity matrix $W \in R^{n \times n}$, $\forall i, j$, $0 \leq W_{ij} \leq 1$ to encode the probability that faces i and j can be clustered in the same path. The spectral analysis of this matrix then suggests a partitioning of the mesh. Liu et al. [2006] use an approximation method [Fowlkes et al. 2004] to improve the performance. Zhang and Liu [2005] adopt a different optimization criterion based on part salience [Hoffman and Singh 1997] for mesh segmentation. Liu and Zhang [2007] use spectral embedding to transform 3D problems to 2D and 1D domains for segmentation.

Implicit Methods. We classify other methods that do not directly extract subregions locally or globally as implicit methods. Two general approaches have been developed in literature for surface decomposition in shape-mapping computation: feature/integral contour tracing methods and skeleton-guided partitioning methods.



Fig. 5. Skeleton-driven decomposition [Yu and Li 2011] and its extension to consistent decomposition.

Feature Contour Tracing. Lee et al. [2005a] trace closed loops that are minimum curvature feature contours on triangle meshes for an intelligent scissoring, and then apply snakes [Kass et al. 1988] to smooth these cutting boundaries. Lévy et al. [2002] seek feature lines via sharp edges of the shape and grow the texture atlas charts inward from these boundaries. Similarly, Mitani and Suzuki [2004] extract feature lines and close them to form subregion boundaries. Wu and Levine [1997] segment the shape by distributing electrical charges on the surface; intuitively, the density charge is very low at concave areas while high at convex areas. The minima rule is followed by tracing contours along regions with local minima of density charge. Edelsbrunner et al. [2001] segment meshes based on the Morse-Smale decomposition, which is extended by Várady et al. [2007] for segmenting CAD models.

Skeleton and Reeb Graph-Based Methods. The skeleton curve concisely encodes the geometry and topology of a 3D object so that it can be used for part decomposition. Li et al. [2001] suggest a partitioning of the object based on skeleton partition. With an extracted skeleton, a plane perpendicular to the skeleton branch sweeps over the mesh and identifies critical points, which then define cuts to segment the mesh to different parts. Berretti et al. [2006] use topological skeletons Biasotti et al. [2003], defined by a geodesic-based mapping function for segmentation. Reniers and Telea [2007] use skeletons to guide the hierarchical mesh segmentation. Raab et al. [2004] extract a voxelized skeleton and partition it to generate an approximated object using bead-like primitives. Lien et al. [2006] simultaneously measure the quality of a centroid or principal axis skeleton and refine the corresponding decomposition together with the guiding skeleton. Yu and Li [2011] compute the guarding skeleton and use it to guide the computation of volumetric star decomposition (Figure 5).

The Reeb graph captures the topology of a geometric region. Based on Morse theory [Milnor 1963], given a n -dimensional compact manifold M , a Reeb graph depicts the relationship among critical points of a smooth real function $f : M \rightarrow \mathcal{R}$ defined on M , so that it reveals the topology of M . A thorough discussion on computations and applications of Reeb graphs is given in Biasotti et al. [2008]. Antini et al. [2005] construct a discrete Reeb graph and construct perceptually meaningful segmentation based on it. The Reeb graph is simplified to cope with oversegmentation. Tierny et al. [2007] use an enhanced Reeb graph based on the harmonic function to guide a hierarchical segmentation procedure. Wang et al. [2009] integrate a Reeb graph (based on harmonic one-form) with topological decomposition to mimic the geometric structure of the shape, in order to get a geometry-adaptive decomposition in which subregions can be parameterized with low distortion.

Consistent Decomposition of Multiple Objects. Shape mapping aims to establish the relationship between decomposed subregions of given objects. Therefore, simultaneously segmenting a set of models into subparts in a coherent manner is often desirable for the divide-and-conquer mapping method. Such a decomposition is referred to as *consistent decomposition*. Specifically, models should be partitioned into the same numbers of subparts. Furthermore, if we construct their dual graphs (each node of the dual graph corresponds to a subpart while each edge of the dual graph connects

each pair of adjacent subparts), the dual graphs should be isomorphic. Intersurface mapping algorithms such as Praun et al. [2001], Schreiner et al. [2004], Kraevoy and Sheffer [2004], and Kwok et al. [2012] trace shortest paths between corresponding feature markers on surfaces to partition both M_1 and M_2 into consistent subregions before mapping each pair of corresponding subregions. The segmentation algorithms in these methods rely on the manually labeled landmarks, and may fail to robustly handle surfaces with nontrivial topology or complex geometry. Shapira et al. [2008] use the normalized Shape-Diameter Function (SDF) [Gal et al. 2007] as a pose-oblivious shape descriptor for consistent segmentation among shapes with similar metric. The iso-contours of the SDF function are traced to separate different parts, with iso-values determined using an Expectation-Maximization (EM) algorithm. The SDF function is suitable to consistently segment surfaces that differ by near-isometry deformation. In Shalom et al. [2008], different shapes are first segmented into parts independently, then a distance measure is created to measure similarities among parts. The distance evaluates both local shape signatures and “the context of the parts within the decomposition” hierarchically. A consistent segmentation is then created based on this catalog of parts with interpart distances. Golovinskiy and Funkhouser [2009] compute the dual graph and apply a clustering based on the inner/interpart connectivity. The segmentation was first used to create a symmetry-respecting segmentation of a single model, then used to transfer segmentations among multiple similar models. Kraevoy et al. [2007] developed a modeling tool that partitions two meshes into subpatches consistently; these corresponding subpatches can be transplanted from one model onto the other. Li et al. [2008b] present a canonical “pants” decomposition algorithm for general surfaces with nontrivial topology. The algorithm partitions a genus- g surface with b boundaries into $2g + b - 2$ canonical topological spheres with three boundaries, called the *pants patches*. When the decomposition is applied on multiple surfaces, the topological coherency of the decomposition is guaranteed, thus it is suitable for the automatic mapping computation between matching high-genus surfaces. The computation of pants decomposition is generalized [Zhang and Li 2012] to better accommodate geometric properties by enumerating homotopic classes in the decomposition space. Consistent shape skeletonization can be used to construct consistent decomposition. Figure 5 shows an example of consistent star decomposition of shapes from consistent guarding skeletons [Yu and Li 2011].

3.2. Local Geometric Features

While global features guide the mapping globally, local geometric features [Tombari et al. 2013], often defined on local points or regions, often effectively indicate potential local correspondence between shapes in mapping computation.

Local curvature and normal information are classic descriptors in 3D object matching [Stein and Medioni 1992; Ruiz-Correa et al. 2001]. These descriptors are typically easy to compute and compare. However, the curvatures and derivatives are very local. Many points possess the same curvature values, and are sensitive to the geometric noise. Dorai and Jain [1997] propose the *Shape-Index* (SI) using the maximum and minimum principal curvatures,

$$SI(\mathbf{x}) = \frac{1}{2} - \frac{1}{\pi} \tan^{-1} \left(\frac{C_M(\mathbf{x}) + C_m(\mathbf{x})}{C_M(\mathbf{x}) - C_m(\mathbf{x})} \right), \quad (4)$$

where $C_M(\mathbf{x})$ and $C_m(\mathbf{x})$ are maximum and minimum principal curvatures at a point $\mathbf{x} \in M$. This is generalized in Chen and Bhanu [2007] to the *Local Surface Patches* (LSP) feature. A salient point is a vertex whose SI value is significantly bigger or smaller than the local average value of its neighboring region. The *Intrinsic Shape Signature* (ISS) feature [Zhong 2009] uses the Eigenvalue decomposition of the scatter

matrix $\Sigma(\mathbf{x})$ of the points in the neighboring regions $N_{\mathbf{x}}$ of a point \mathbf{x} ,

$$\Sigma(\mathbf{x}) = \frac{1}{N} \sum_{\mathbf{p} \in N_{\mathbf{x}}} (\mathbf{p} - \mu_{\mathbf{x}})(\mathbf{p} - \mu_{\mathbf{x}})^T, \quad (5)$$

where $\mu_{\mathbf{x}} = \frac{\sum_{\mathbf{p} \in N_{\mathbf{x}}} \mathbf{p}}{N}$. The *KeyPoint Quality* (KPQ) feature [Mian et al. 2010] extracts feature points by analyzing the first two principal eigenvalues of $\Sigma(\mathbf{x})$. The *slippage* feature [Bokeloh et al. 2008] extracts regions that have the maximal stability under the rigid matching of the surface with itself. A few image descriptors [Lowe 2004; Mikolajczyk and Schmid 2005] can be effectively generalized to surfaces. For example, the *3D Harris* descriptor [Sipiran and Bustos 2010] extends the well-known Harris corner detector in image processing to surfaces. The algorithm fits a vertex's local neighboring region using a quadratic patch, which is treated as an image. After Gaussian smoothing, the derivatives are used to identify geometric corners/edges.

3.3. Multiscale Shape Features

Multiscale shape features first build a scale space, then extract extremals across different scales. The multiscale feature detectors are widely used in describing 2D images [Lindeberg 1998]. In 3D shape mapping, one can either extend the scale-space theory to 3D shapes or parameterize the 3D shapes onto planar domains, then apply the multiscale analysis in 2D. We recap a few popular multiscale features as follows.

- Laplace-Beltrami Scale-space*. Unnikrishnan and Hebert [2008] build a scale space using the Laplace-Beltrami operator Δ_M , employing increasing support regions around points on the 3D mesh.
- Heat Kernel*. Closely related to Laplace-Beltrami, the Heat Kernel function follows the heat diffusion process to describe geometric information around a point. Specifically, the dissipating time t of heat is an intrinsic scale to describe the neighborhood of a point. Local shape properties are encoded by this function within a small time range, while global properties are encoded in a larger time range. The kernel function of solutions to the heat diffusion on the shape is used as the *heat kernel signature* (HKS) [Sun et al. 2009]. Using HKS, salient points are extracted as the local maxima of the neighborhood within a certain range of time t . This method can detect a very limited number of keypoints that are highly repeatable and isometry invariant.
- MeshDoG*. The MeshDoG feature builds a scale-space using the difference of Gaussian (DoG) operator on a scalar function of the mesh. The scalar function can either be extrinsic like coordinates [Castellani et al. 2008] or intrinsic such as mean curvature or Gaussian curvature [Zaharescu et al. 2009]. Geometric saliency is determined within its neighborhood on the current and adjacent scales.
- Normal / Curvature Map of the Flattened Surface*. Akagunduz and Ulusoy [2007] and Novatnack and Nishino [2008] construct the scale space using the 2D parameterization of the 3D surface. The flattened geometry is resampled regularly; upon the flat region the interpolated normal map and mean/Gaussian curvature maps are used for saliency detection.
- 3D SURF*. Knopp et al. [2010] builds a scalespace for surface M using its voxelization. With this representation, box filtering can be computed efficiently for saliency extraction. Saliency is measured using the Hessian of second-order Gaussian derivatives and computed for each grid bin and for each octave.
- Mesh Saliency*. Lee et al. [2005b] define the *mesh saliency* detector using a perception-based metric that extract regions that are different from their surrounding context. The descriptor uses the Gaussian-weighted enter-surround evaluation of surface curvatures.

3.4. Solving Feature Correspondence

When geometric features of shapes are extracted and described using descriptors, their correspondence can be formulated as an assignment problem on a graph. Let $P = \{p_1, p_2, \dots, p_{n_1}\}$ and $Q = \{q_1, q_2, \dots, q_{n_2}\}$ be the sets of features on objects M_1 and M_2 , respectively. An optimal assignment \hat{f} maps a feature on the source object to a feature on the target object, so that it maximizes the following matching score:

$$\hat{f} = \operatorname{argmax}_f \left(\sum_{p \in P} c(p, f(p)) + \lambda \sum_{(p, p') \in P} g((p, p'), (f(p), f(p'))) \right), \quad (6)$$

where $c(p, q)$ is a cost function measuring feature similarity between features p and q , and $g((p, p'), (q, q'))$ is another function measuring the inherent mutual consistency between the assignment (p, q) and assignment (p', q') . If one uses an association graph G to encode feature correspondence, where each node of G is a potential assignment $(p, q) \in P \times Q$ and each edge of G indicates a pair of assignments $((p, q), (p', q'))$, whose mutual consistency can be measured by function g , then these similarity measures c and g define weights on nodes and edges of G , respectively. The solving of optimal assignment is then often formulated as an integer quadratic program (IQP) that finds a permutation matrix (or its column vector replica) $\hat{\mathbf{x}}$ such that:

$$\begin{aligned} \hat{\mathbf{x}} &= \operatorname{argmax}_{\mathbf{x}} (\mathbf{x}^T M \mathbf{x}) \\ \text{s.t. } \mathbf{x} &\in \{0, 1\}^{n_1 n_2}, \forall i \sum_{a=1}^{n_2} x_{ia} \leq 1, \forall a \sum_{i=1}^{n_1} x_{ia} \leq 1, \end{aligned} \quad (7)$$

where element $x_{ia} = 1$ indicates that p_i is mapped to q_a and $x_{ia} = 0$ indicates otherwise, and the second and third constraints enforce the assignment to be one to one. This IQP algorithm is NP-complete, and effectively finding its general approximate solutions have been proposed in many fields of computer science research. A few classic algorithms, such as the Hungarian algorithm [Munkres 1957], voting algorithm [Lowe 2004], or RANSAC [Fischler and Bolles 1981] algorithm, despite their effectiveness in many applications, do not actually solve this optimization problem. More recently, algorithms that seek for better approximate solution to this IQP problem have been more systematically studied. For example, Leordeanu and Hebert [2005] suggest a spectral relaxation algorithm by greedily selecting the leading eigenvector of the affinity matrix M . Beg et al. [2005] approximate the quadratic function using linear programming problems, which were solved through local gradient descent approaches. More detailed discussion on solving IQP problems can be found in Loiola et al. [2007]. Although IQP is a combinatorial integer problem, an effective approach is to relax its integer constraint and solve it as a continuous optimization problem (to be discussed in Section 4), then project/round such an approximate solution to the space defined by the integer and linear bound constraints.

4. OPTIMIZATION METHODS

With the objective functions and constraints formulated in mapping computation, effective numerical optimization algorithms are needed to solve them. We classify the optimization methods into *deterministic* and *stochastic* strategies.

4.1. Deterministic Optimization Algorithms

Given the nonlinear optimization problem

$$\hat{\mathbf{x}} = \operatorname{argmin}_{\mathbf{x}} E(\mathbf{x}), \quad (8)$$

we want to find the solution $\hat{\mathbf{x}}$ that minimizes the objective function $E(\mathbf{x})$ subject to certain constraints. We can classify deterministic numerical algorithms following the orders of derivatives used in optimization: (1) zero-order or derivative-free algorithms; (2) first-order algorithms, such as gradient descent or conjugate gradients approaches; and (3) second-order algorithms, such as Newton methods. Generally, an iterative strategy is employed to optimize the variable \mathbf{x} :

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k, k = 0, 1, \dots, K, \quad (9)$$

where \mathbf{d}_k is the search direction at k -th iteration and λ_k is the scalar parameter that determines the step size along the search direction. Different λ_k and \mathbf{d}_k have been designed to make the sequence $\{\mathbf{x}_k\}$ to converge to a (local) minimum of the objective function as fast as possible.

4.1.1. Zero-order or Derivative-Free Algorithms. We first consider optimization algorithms that do not require derivatives to be explicitly provided. Typical algorithms of this type include various combinatorial enumerations [Loiola et al. 2007] and greedy selection [Leordeanu and Hebert 2005]. In many such problems, objective functions are not smooth and derivatives are unavailable. Sometimes, derivatives do exist but are too expensive to evaluate. Derivative-free approaches that approximate gradients locally without explicitly evaluating them could be effective. For example, in Wan et al. [2011], the mapping distortion needs to be measured after composing a few discrete tracings of integral lines, causing derivative information to be unavailable. A derivative-free optimization algorithm based on active-set gradient estimation is used to solve the nonlinear optimization problem.

4.1.2. Gradient Descent and Conjugate Gradients Methods. Objective functions with gradient information available can usually be solved efficiently using first-order derivatives. The standard *gradient descent* (GD) approach reduces the function value $E(\mathbf{x})$ by moving toward the negative of the gradient, $-\lambda \nabla E(\mathbf{x})$, that is,

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \lambda_k \mathbf{g}(\mathbf{x}_k), \quad (10)$$

where $\mathbf{g}(\mathbf{x}_k)$ is the derivative of E and λ_k is either some constant, decaying function of k , or obtained by a line search.

The *conjugate gradient* (CG) method is usually more efficient than GD. The search direction \mathbf{d}_k is designed as a linear combination of the previous search direction \mathbf{d}_{k-1} and the current gradient $\mathbf{g}(\mathbf{x}_k)$:

$$\mathbf{d}_k = -\mathbf{g}_k + \beta \mathbf{d}_{k-1}, \quad (11)$$

where $\mathbf{d}_0 = -\mathbf{g}_0$. Different choices of scalar β can influence how the optimization converges globally.

4.1.3. Newton, Quasi-Newton, L-M Methods. The classic Newton method is a second-order optimization algorithm:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [H(\mathbf{x}_k)]^{-1} \mathbf{g}(\mathbf{x}_k), \quad (12)$$

where $H(\mathbf{x}_k)$ is the Hessian matrix of the objective function at \mathbf{x}_k . This second-order information improves the convergence of the optimization algorithm. But the computation of the Hessian matrix and its inverse can be computationally expensive, especially for large amounts of data.

The Quasi-Newton methods, in contrast, use an approximation to the inverse of the Hessian, $L_k \approx [H(\mathbf{x}_k)]^{-1}$. During each iteration, the update actually uses only the first-order derivatives, without computing the second-order derivatives. In this sense, rigorously speaking the Quasi-Newton method is a first-order algorithm. The step-size

λ_k is determined by an inexact line search:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \lambda_k L_k \mathbf{g}(\mathbf{x}_k). \quad (13)$$

With certain conditions, Quasi-Newton methods often have superlinear convergence. Many methods to construct the series L_k have been developed. Notable ones include the Symmetric-Rank-1 (SR1), Davidon-Fletcher-Powell (DFP), and Broyden-Fletcher-Goldfarb-Shanno (BFGS). The BFGS is widely used and demonstrates very good numerical efficiency in many applications. Its update rule is

$$L_{k+1} = \left(I - \frac{\mathbf{s}_k \mathbf{y}_k^T}{\mathbf{s}_k^T \mathbf{y}_k} \right) L_k \left(I - \frac{\mathbf{y}_k \mathbf{s}_k^T}{\mathbf{s}_k^T \mathbf{y}_k} \right) + \frac{\mathbf{s}_k \mathbf{s}_k^T}{\mathbf{s}_k^T \mathbf{y}_k}, \quad (14)$$

where I is the identity matrix, $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$, and $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$. The step-size λ_k is determined by the strong Wolfe conditions [Moré and Thuente 1994]. The limited-memory version of BFGS method, LBFGS, is a practical variant of BFGS. It avoids storing the matrix L_k in memory, resulting in great efficiency for large optimization problems. But the LBFGS method sometimes requires a larger number of function evaluations, and may be inefficient when the problem is ill-conditioned (e.g., when the Hessian matrix contains a wide distribution of eigenvalues). Another limitation of LBFGS is that it needs accurate gradient evaluation, which is sometimes not easy when mapping shapes with noise geometry or material.

4.1.4. Deterministic Approaches to Overcome Local Minima. When the nonlinear objective function is nonconvex, deterministic optimization often converges to local minima. A few approaches have been developed to overcome local minima and obtain a desirable global solution.

Branch and Bound and Tree-based Search. A branch-and-bound algorithm first starts with the original problem (root problem) with the complete feasible region. It terminates if the solution matches the estimated lower bound; otherwise, it partitions the feasible region into subregions and recursively solves the subproblems. Each subproblem's solution is also a solution (not necessarily a global optimum) to the original problem, and is used to prune the expansion of the partitioning tree: branches with worse lower bound than an existing solution need not be expanded further. The searching proceeds until all nodes are examined or until the existing solution is within a threshold range of the lower bounds of all unexpanded nodes. Through branch and bound, feature modeling or shape mapping algorithms reduced to integer [Yu and Li 2011] or mix-integer [Bommès et al. 2009; Yu et al. 2013] problems are converted to continuous problems to get good approximate solutions before being rounded and reoptimized. *Beam search* is another tree search approach to reduce the exponential expansion of branches. For example, McBride and Kimia [2003] grow each node up to only k children nodes each time and preserve at most m branches in each level in their tree search.

Graph-based Search. To determine the optimal combinatorial or integer variables often requires a thorough search on some structure/state graphs. A state graph is used to represent the combinatorial states of different variables where each node indicates a configuration. The optimization then explores all reachable nodes and seeks for the best solution. Zhang and Li [2012] enumerate pants decompositions of different homotopic classes on the decomposition's dual graph to obtain an optimal partitioning. Huber [2002] registers multiple-range scans simultaneously by growing a minimum spanning tree on a configuration graph, where each edge represents a pairwise alignment. Huang et al. [2006] and Zhang and Li [2014] refine a groupwise partial matching by finding a "maximal compatible edge set" on a graph, which improves the initial solution computed by iterative pairwise matching.

Hierarchical Optimization. Solving optimization in a hierarchical scheme is a traditional approach to circumvent local minima. Geometries can be tessellated hierarchically so that their mapping can be first solved using coarser resolutions, then refined in denser resolutions. These strategies have been adopted in regular mesh structure (multiresolutional grids [Metz et al. 2011]) and irregular mesh structures (progressive meshes [Schreiner et al. 2004; Kwok et al. 2012; Wan et al. 2012, 2013]). Another type of hierarchical algorithm is conducted on frequency domain. Shapes are described using different coefficients, where low-frequency coefficients encode the shape’s rough geometry and high-frequency coefficients capture the geometric details. Spectrum-based matching techniques such as Laplacian eigenfunctions [Jain et al. 2007; Mateus et al. 2008], diffusion kernel [Bronstein et al. 2010], manifold harmonics [Zhong et al. 2012; Vallet and Lévy 2008], Zernike decomposition [Novotni and Klein 2003], and so forth, have been used in various 3D shape mapping and registration problems.

Function Convexification. Shape mapping generally reduces to minimizing a nonconvex function $E(\mathbf{x})$. A convex function $E'(\mathbf{x}) = E(\mathbf{x}) + \beta E_R(\mathbf{x})$ can be used to approximate $E(\mathbf{x})$, where $E_R(\mathbf{x})$ is the regularization term and β is the weighting factor that balances the convexity of E' and its approximation accuracy to E . Chui and Rangarajan [2003] model transformations using thin plate splines and employ graduated assignment [Gold and Rangarajan 1996] for nonrigid shape registration. Liu [2007] uses a mean field annealing algorithm, which is a deterministic approximation to simulated annealing, in solving nonrigid shape matching.

4.2. Stochastic Optimization Algorithms

When the geometric data have many noises, outliers, or local optima, statistics or probabilistic approaches can be used to develop more robust optimization. Similarly, we classify the stochastic optimization algorithms through the derivative information they require: (1) voting and RANSAC algorithms; (2) stochastic gradient descent and conjugate gradient algorithms. Second-order derivatives are less used in stochastic optimization, probably due to their sensitivity to noise and outliers in addition to their expensive calculations.

4.2.1. Voting and RANSAC. The voting approaches [Li and Guskov 2005; Lipman and Funkhouser 2009] accumulate the effects of choosing different parameters to determine the optimal variables in the optimization. Random Sample Consensus (RANSAC) is another efficient algorithm for fitting parameters when the problem encounters many outliers [Tevs et al. 2011, 2009]. Anguelov et al. [2005] used Markov random field to maximize the joint probability of correspondence in nonrigid shape registration.

4.2.2. Stochastic GD and Stochastic CG Algorithms. The stochastic gradient descent (GD) method [Kushner and Yin 2003] can be considered as a statistical variant of the deterministic GD algorithm. It determines the step-size λ_k and search direction d_k iteratively as follows:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k - \gamma_k \tilde{g}_k, \\ \tilde{g}_k &= g(\mathbf{x}_k) + \varepsilon_k, \end{aligned} \quad (15)$$

where \tilde{g}_k denotes an approximation of the true derivative g_k at \mathbf{x}_k , and ε_k is the approximation error. If $\varepsilon_k = 0$, this becomes the deterministic GD procedure. During each iteration, $g(\mathbf{x}_k)$ is approximated by calculating only the derivative components on a small randomly sampled element. According to Kushner and Yin [2003], if the bias of the expected approximation error goes to zero, namely, $Exp(\tilde{g}_k) = g(\mathbf{x}_k)$, the convergence of this stochastic optimization to the true solution $\hat{\mathbf{x}}$ can be guaranteed. Such stochastic GD methods are effective when exact gradient information is not available, not reliable, or too expensive. In these cases, using this statistically approximated gradient

could improve the algorithm's efficiency and stability against noise. The stochastic GD strategy can also be revised to a stochastic conjugate gradient (CG) approach [Xu et al. 2013b], with better convergence rates. Stochastic GD and CG algorithms are effective for inhomogeneous volume mapping between images whose intensity information (data fitting term) is noisy [Metz et al. 2011; Xu and Li 2013b].

4.3. Summary and Discussion on Optimization Strategies

Deterministic algorithms iteratively optimize the search direction and step size. When function and derivative evaluations are available and reliable, they yield efficient and stable convergence. When such evaluations are prohibitive (bad accuracy due to noise or bad efficiency in modeling large geometries), stochastic algorithms are effective alternative strategies. The efficiency of stochastic optimization compared with the deterministic optimization, however, is debatable: although stochastic optimization has significantly reduced derivative estimation/approximation time, it often requires more iterations to converge.

5. SUMMARY AND CONCLUSION

We studied the cross-shape mapping problem to find a bijective map between given shapes minimizing specific distortions. We classified and compared different mapping algorithms by considering them as different optimization problems with specific objective functions, geometric constraints, and solving schemes.

The automatic computation of such correspondence between arbitrarily given shapes with nontrivial topology and geometry remains a challenging problem. Directly solving the global intershape map often inevitably gets trapped in local minima under most existing intrinsic or extrinsic matching metrics. Therefore, most mapping algorithms utilize various shape features to guide the optimization and avoid undesirable local optima. Many advanced numerical optimization strategies have also been developed for efficient mapping computation. Divide-and-conquer could be another global strategy to solve this problem, which partitions the complex shapes into solvable subparts. However, effective consistent decomposition of multiple shapes into compatible sets of subparts with simple topology also remains challenging.

Volumetric data mapping and registration can benefit the analysis of heterogeneous geometric data. However, with more complicated topology and interior structure for 3-manifolds, the research in this topic has just started. The existence of bijective volumetric parameterization for even trivial topological domains is generally unknown. Domain decomposition techniques that partition volume regions into simple subregions could be a solution, but research on consistent volume decomposition and mapping computation has just begun.

Acquired 3D data usually carry various topological and geometric noises. Developing intershape matching algorithms for noisy data has not been thoroughly discussed in this article but is important in many real tasks. Modeling and aligning features that are reliable against noise can facilitate the mapping of noisy data. Hence, it is an active research area in computer vision.

REFERENCES

- A. Agathos, I. Pratikakis, S. Perantonis, N. Sapidis, and P. Azariadis. 2007. 3D mesh segmentation methodologies for CAD applications. *Comput. Aided Design* 4, 6, 827–841.
- E. Akagunduz and I. Ulusoy. 2007. 3D object representation using transform and scale invariant 3D features. In *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV'07)*. 1–8.
- M. Alexa. 1999. Merging polyhedral shapes with scattered features. In *Proceedings of the International Conference on Shape Modeling and Applications*. 202–210.

- B. Allen, B. Curless, and Z. Popović. 2003. The space of human body shapes: Reconstruction and parameterization from range scans. In *ACM SIGGRAPH*. 587–594.
- P. Alliez, É. C. deVerdière, O. Devillers, and M. Isenburg. 2003. Isotropic surface remeshing. In *Proceedings of Shape Modeling International*. 49–58.
- B. Amberg, S. Romdhani, and T. Vetter. 2007. Optimal step nonrigid ICP algorithms for surface registration. In *CVPR*. IEEE Computer Society, 1–8.
- D. Anguelov, D. Koller, P. Srinivasan, and H.C. Pang. 2005. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. *NIPS* 17, 33–40.
- G. Antini, S. Berretti, A. Bimbo, and P. Pala. 2005. 3D mesh partitioning for retrieval by parts applications. In *ICME*. 1210–1213.
- K. S. Arun, T. S. Huang, and S. D. Blostein. 1987. Least-squares fitting of two 3-D point sets. *IEEE Trans. Pattern Anal. Mach. Intell.* 9, 5, 698–700.
- A. Asirvatham, E. Praun, and H. Hoppe. 2005. Consistent spherical parameterization. In *Proceedings of the International Conference on Computational Science (2)*. 265–272.
- D. Attali, J. D. Boissonnat, and H. Edelsbrunner. 2009. Stability and computation of medial axes—a state-of-the-art report. In *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*. Springer, New York, NY.
- M. Attene, B. Falcidieno, and M. Spagnuolo. 2006. Hierarchical mesh segmentation based on fitting primitives. *Vis. Comput.* 22, 3, 181–193.
- D. Ballard. 1981. Generalized hough transform to detect arbitrary patterns. *IEEE Trans. Patt. Anal. Mach. Intell.* 13, 2, 111–122.
- M. F. Beg, M. I. Miller, A. Troune, and L. Younes. 2005. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *Int. J. Comput. Vision* 61, 2.
- P. Benkó and T. Várady. 2002. Direct segmentation of smooth, multiple point regions. In *Geometric Modeling and Processing—Theory and Applications*. 169.
- J. Bennett, V. Pascucci, and K. Joy. 2008. A genus oblivious approach to cross parameterization. *Comput. Aided Geom. Des.* 25, 8, 592–606.
- S. Berretti, A. Del Bimbo, and P. Pala. 2006. Partitioning of 3D meshes using Reeb graph. In *Proceedings of the 18th International Conference on Pattern Recognition*. 19–22.
- P. J. Besl and N. D. McKay. 1992. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* 14, 2, 239–256.
- S. Biasotti, L. De Floriani, B. Falcidieno, P. Frosini, D. Giorgi, C. Landi, L. Papaleo, and M. Spagnuolo. 2008. Describing shapes by geometrical-topological properties of real functions. *ACM Comput. Surv.* 40, 4, 1–87.
- S. Biasotti, S. Marini, M. Mortara, and G. Patané. 2003. An overview on properties and efficacy of topological skeletons in shape modelling. In *Shape Modeling International*. 245–256, 297.
- H. Biermann, I. Martin, F. Bernardini, and D. Zorin. 2002. Cut-and-paste editing of multiresolution surfaces. *ACM Trans. Graphics* 21, 3, 312–321.
- H. Blum. 1967. A transformation for extracting new descriptors of shape. In *Models for the Perception of Speech and Visual Form*. MIT Press, Cambridge, MA, 362–380.
- I. Boier-Martin, H. Rushmeier, and J. Jin. 2004. Parameterization of triangle meshes over quadrilateral domains. In *Proceedings of the Symposium on Geometry Processing*. 193–203.
- M. Bokeloh, A. Berner, M. Wand, H.-P. Seidel, and A. Schilling. 2008. Slippage Features. Wilhelm-Schickard-Institut, Universität Tübingen, Tübingen, Germany, 1–20.
- D. Bommers, M. Campen, H.-C. Ebke, P. Alliez, and L. Kobbelt. 2013. Integer-grid maps for reliable quad meshing. *ACM Trans. Graph.* 32, 4, 98.
- D. Bommers, H. Zimmer, and L. Kobbelt. 2009. Mixed-integer quadrangulation. *ACM Trans. Graph.* 28, 3, 77:1–77:10.
- F. Bookstein. 1989. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Mach. Intell.* 11, 6, 567–585.
- G. Borgefors. 1996. On digital distance transforms in three dimensions. *Comput. Vis. Image Underst.* 64, 3, 368–376.
- P. L. Bowers and M. K. Hurdal. 2003. Planar conformal mappings of piecewise flat surfaces. In *Visualization and Mathematics III*. 3–34.
- A. Bronstein, M. Bronstein, and R. Kimmel. 2008. *Numerical Geometry of Non-Rigid Shapes*. Springer, New York, NY.

- A. M. Bronstein, M. M. Bronstein, and R. Kimmel. 2007. Calculus of non-rigid surfaces for geometry and texture manipulation. *IEEE Trans. Visual. Comput. Graph.* 902–913.
- A. M. Bronstein, M. M. Bronstein, R. Kimmel, M. Mahmoudi, and G. Sapiro. 2010. A Gromov-Hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching. *Int. J. Comput. Vision* 89, 2–3, 266–286.
- M. Bronstein and I. Kokkinos. 2010. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*. 1704–1711.
- B. Brown and S. Rusinkiewicz. 2007. Global non-rigid alignment of 3-D scans. *ACM Trans. Graph. (Proc. SIGGRAPH)* 26, 3, 21:1–21:9.
- L. Brown. 1992. A survey of image registration techniques. *Comput. Surveys* 24, 325–376.
- J. Cao, X. Li, Z. Chen, and H. Qin. 2012. Spherical DCB-spline surfaces with hierarchical and adaptive knot insertion. *IEEE Trans. Visual. Comput. Graphics* 18, 8, 1290–1303.
- J. Cao, X. Li, G. Wang, and H. Qin. 2009. Surface reconstruction using bivariate simplex splines on Delaunay configurations. *Comput. Graphics* 33, 3, 341–350.
- U. Castellani, M. Cristani, S. Fantoni, and V. Murino. 2008. Sparse points matching by combining 3D mesh saliency with statistical descriptors. *Comput. Graphics Forum* 27, 2, 643–652.
- B. Chazelle, D. P. Dobkin, N. Shouraboura, and A. Tal. 1997. Strategies for polyhedral surface decomposition: An experimental study. *Comput. Geom. Theory Appl.* 7, 5–6, 327–342.
- H. Chen and B. Bhanu. 2007. 3D free-form object recognition in range images using local surface patches. *Pattern Recogn. Lett.* 28, 10, 1252–1262.
- X. Chen, A. Golovinskiy, and T. Funkhouser. 2009. A benchmark for 3D mesh segmentation. *ACM Trans. Graphics (Proc. SIGGRAPH)* 28, 3, 1–12.
- G. Christensen, R. Rabbitt, and M. Miller. 1996. Deformable templates using large deformation kinematics. *IEEE Trans. Image Process.* 5, 10, 1435–1447.
- J. Chuang, N. Ahuja, C. Lin, C. Tsai, and C. Chen. 2004. A potential-based generalized cylinder representation. *Comput. Graphics* 28, 6, 907–918.
- H. Chui and A. Rangarajan. 2003. A new point matching algorithm for non-rigid registration. *Comput. Vis. Image Underst.* 89, 2–3, 114–141.
- R. R. Coifman and S. Lafon. 2006. Diffusion maps. *Appl. Comput. Harmonic Anal.* 21, (2006), 5–30.
- C. R. Collins and K. Stephenson. 2003. A circle packing algorithm. *Comput. Geom. Theory Appl.* 25, 3, 233–256.
- N. D. Cornea, D. Silver, and P. Min. 2007. Curve-skeleton properties, applications, and algorithms. *IEEE Trans. Visual. Comput. Graphics* 13, 3, 530–548.
- D. DeCarlo and J. Gallier. 1996. Topological evolution of surfaces. In *Proceedings of Graphics Interface*. 194–203.
- P. Degener, J. Meseth, and R. Klein. 2003. An adaptable surface parameterization method. In *Proceedings of the 12th International Meshing Roundtable*. 201–213.
- M. Desbrun, M. Meyer, and P. Alliez. 2002. Intrinsic parameterizations of surface meshes. *Comput. Graphics Forum* 21, 3, 209–218.
- T. Dey, K. Li, and J. Sun. 2007. On computing handle and tunnel loops. In *IEEE Proc. NASAGEM 07*. 357–366.
- T. Dey and J. Sun. 2006. Defining and computing curve-skeletons with medial geodesic function. In *Proceedings of the Eurographics Symposium on Geometry Processing*. 143–152.
- T. K. Dey and W. Zhao. 2002. Approximate medial axis as a voronoi subcomplex. In *Proceedings of the ACM Symposium on Solid Modeling and Applications*. 356–366.
- A. Dominitz and A. Tannenbaum. 2010. Texture mapping via optimal mass transport. *IEEE Trans. Vis. Comput. Graphics* 16, 3, 419–433.
- G. Donato and S. Belongie. 2002. Approximate thin plate spline mappings. In *Proc. ECCV*. 21–31.
- S. Dong, S. Kircher, and M. Garland. 2005. Harmonic functions for quadrilateral remeshing of arbitrary manifolds. *Comput. Aided Geom. Des.* 22, 5, 392–423.
- C. Dorai and A. Jain. 1997. COSMOS-A representation scheme for 3D free-form objects. *IEEE Trans. Pattern Anal. Mach. Intell.* 19, 10, 1115–1130.
- M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. 1995. Multiresolution analysis of arbitrary meshes. In *SIGGRAPH*. 173–182.
- H. Edelsbrunner, J. Harer, and A. Zomorodian. 2001. Hierarchical morse complexes for piecewise linear 2-manifolds. In *Proceedings of the Symposium on Computational Geometry*. 70–79.

- D. W. Eggert, A. Lorusso, and R. B. Fisher. 1997. Estimating 3-D rigid body transformations: A comparison of four major algorithms. *Mach. Vision Appl.* 9, 5–6, 272–290.
- A. Elad and R. Kimmel. 2003. On bending invariant signatures for surfaces. *IEEE Trans. Pattern Anal. Mach. Intell.* 25, 10, 1285–1295.
- G. Farin. 2002. *Curves and Surfaces for CAGD*. Morgan-Kaufmann, San Francisco, CA.
- V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. 2008. Groups of adjacent contour segments for object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 30, 1, 36–51.
- M. A. Fischler and R. C. Bolles. 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 6, 381–395.
- M. Floater. 1997. Parameterization and smooth approximation of surface triangulations. *Comput. Aided Geom. Des.* 14, 3, 231–250.
- M. Floater. 2003. Mean value coordinates. *Comput. Aided Geom. Des.* 20, 1, 19–27.
- M. Floater and K. Hormann. 2005. Surface parameterization: A tutorial and survey. In *Advances in Multiresolution for Geometric Modelling*. 157–186.
- M. Fornefett, K. Rohr, and H. S. Stiehl. 2001. Radial basis functions with compact support for elastic registration of medical images. *Image Vision Comput.* 19, 1–2, 87–96.
- F. Fouss, A. Pirrotte, J.-M. Renders, and M. Saerens. 2007. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Trans. Knowl. Data Eng.* 19, 3 (2007), 355–369.
- C. Fowlkes, S. Belongie, F. Chung, and J. Malik. 2004. Spectral grouping using the Nyström method. *IEEE Trans. Pattern Anal. Mach. Intell.* 26, 2, 214–225.
- I. Friedel, P. Schröder, and M. Desbrun. 2005. Unconstrained spherical parameterization. In *ACM SIGGRAPH*. 134.
- R. Gal, A. Shamir, and D. Cohen-Or. 2007. Pose-oblivious shape signature. *IEEE Trans. Vis. Comput. Graph.* 13, 2, 261–271.
- M. Garland, A. Willmott, and P. S. Heckbert. 2001. Hierarchical face clustering on polygonal surfaces. In *Proceedings of the Symposium on Interactive 3D*. 49–58.
- N. Gelfand and L. J. Guibas. 2004. Shape segmentation using local slippage analysis. In *Proceedings of the Symposium on Geometry Processing*. 214–223.
- N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann. 2005. Robust global registration. In *Proceedings of the Symposium on Geometry Processing*. 197–206.
- J. Glaunes, A. Qiu, M. Miller, and L. Younes. 2008. Large deformation diffeomorphic metric curve mapping. *Int. J. Comput. Vision* 80, 3, 317–336.
- J. Glaunes, M. Vaillant, and M. I. Miller. 2004. Landmark matching via large deformation diffeomorphisms on the sphere. *J. Math. Imag. Vision* 20, 1–2, 179–200.
- S. Gold and A. Rangarajan. 1996. A graduated assignment algorithm for graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.* 18, 4, 377–388.
- A. Golovinskiy and T. Funkhouser. 2009. Consistent segmentation of 3D models. *Comput. Graphics* 33, 3, 262–269.
- W. Gong and G. Bertrand. 1990. A simple parallel 3D thinning algorithm. In *Proceedings of the 10th International Conference on Pattern Recognition*. 188–190.
- S. Gortler, C. Gotsman, and D. Thurstun. 2006. Discrete one-forms on meshes and applications to 3D mesh parameterization. *Comput. Aided Geom. Des.* 23, 2, 83–112.
- C. Gotsman, X. Gu, and A. Sheffer. 2003. Fundamentals of spherical parameterization for 3D meshes. *ACM Trans. Graph.* 22, 3, 358–363.
- L. Grady. 2006. Random walks for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 28, 11, 1768–1783.
- A. Gregory, A. State, M. Lin, D. Manocha, and M. Livingston. 1998. Feature-based surface decomposition for correspondence and morphing between polyhedra. In *Proceedings of Computer Animation*. 64–71.
- J. Gregson, A. Sheffer, and E. Zhang. 2011. All-hex mesh generation via volumetric polycube deformation. *Comput. Graphics Forum* 2011 30, 5.
- C. Grimm and J. Hughes. 2003. Parameterizing n-holed tori. In *Mathematics of Surfaces X*. 14–29.
- X. Gu and S.-T. Yau. 2003. Global conformal surface parameterization. In *Proceedings of the Symposium on Geometry Processing*. 127–137.
- X. Gu and S. T. Yau. 2007. *Computational Conformal Geometry*. International Press, Somerville, MA.
- I. Guskov. 2004. An anisotropic mesh parameterization scheme. *Eng. Comput.* 20, 2, 129–135.

- S. Haker, S. Angenent, A. Tannenbaum, R. Kikinis, G. Sapiro, and M. Halle. 2000. Conformal surface parameterization for texture mapping. *IEEE Trans. Vis. Comput. Graph.* 6, 2, 181–189.
- S. Han, J. Xia, and Y. He. 2010. Hexahedral shell mesh construction via volumetric polycube map. In *Proceedings of the ACM Symposium on Solid and Physical Modeling*. 127–136.
- A. Hatcher. 2002. *Algebraic Topology*. Cambridge University Press, New York, NY.
- Y. He, H. Wang, C.-W. Fu, and H. Qin. 2009a. A divide-and-conquer approach for automatic polycube map construction. *Comput. Graph.* 33, 3, 369–380.
- Y. He, X. Xiao, and H.-S. Seah. 2009b. Harmonic 1-form based skeleton extraction from examples. *Graph. Models* 71, 2, 49–62.
- Y. Hecker and R. Bolle. 1994. On geometric hashing and the generalized hough transform. *IEEE SMC* 24, 9, 1328–1338.
- M. Held, J. Klosowski, and J. Mitchell. 1996. Real-time collision detection for motion simulation within complex environments. In *ACM SIGGRAPH '96*. 151.
- M. Hilaga, Y. Shinagawa, T. Kohmura, and T. Kunii. 2001. Topology matching for fully automatic similarity estimation of 3D shapes. In *SIGGRAPH*. 203–212.
- D. Hoffman, W. Richards, A. Pentl, J. Rubin, and J. Scheuhammer. 1984. Parts of recognition. *Cognition* 18, 65–96.
- D. Hoffman and M. Singh. 1997. Saliency of visual parts. *Cognition* 63, 1, 29–78.
- K. Hormann and G. Greiner. 2000. MIPS: An efficient global parameterization method. In *Curve and Surface Design*. 153–162.
- K. Hormann, B. Lévy, and A. Sheffer. 2007. Mesh parameterization: Theory and practice. In *ACM Siggraph 2007 Course*, Vol. 11, 1–87.
- B. Horn. 1987. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am.* 4, 4 (1987), 629–642.
- B. Horn, H. Hilden, and S. Negahdaripour. 1988. Closed-form solution of absolute orientation using orthonormal matrices. *J. Opt. Soc. Am.* 5, 7, 1127–1135.
- T. Hou and H. Qin. 2010. Efficient computation of scale-space features for deformable shape correspondences. In *ECCV*. 384–397.
- H. Huang, L. Shen, R. Zhang, F. Makedon, B. Hettleman, and J. Pearlman. 2005. Surface alignment of 3D spherical harmonic models: application to cardiac MRI analysis. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Interventions*. 67–74.
- J. Huang, Y. Tong, H. Wei, and H. Bao. 2011. Boundary aligned smooth 3D cross-frame field. *ACM Trans. Graph.* 30, 6, 143.
- Q.-X. Huang, S. Flöry, N. Gelfand, M. Hofer, and H. Pottmann. 2006. Reassembling fractured objects by geometric matching. In *Proceedings of SIGGRAPH'06*. ACM, 569–578.
- D. F. Huber. 2002. *Automatic three-dimensional modeling from reality*. Ph.D. dissertation. Carnegie Mellon University, Pittsburgh, PA, CMU-RI-TR-02-35.
- K. Inoue, I. Takayuki, Y. Atsushi, F. Tomotake, and S. Kenji. 2001. Face clustering of a large-scale CAD model for surface mesh generation. *Comput. Aided Des.* 33, 251–261.
- M. Isenburg, S. Gumhold, and C. Gotsman. 2001. Connectivity shapes. In *IEEE Visualization*. 135–142.
- S. S. Iyengar, X. Li, H. Xu, S. Mukhopadhyay, N. Balakrishnan, A. Sawant, and P. Iyengar. 2012. Toward more precise radiotherapy treatment of lung tumors. *IEEE Computer* 45, 59–65.
- V. Jain and H. Zhang. 2006. Robust 3D shape correspondence in the spectral domain. In *SMI*. 19.
- V. Jain, H. Zhang, and O. Kaick. 2007. Non-rigid spectral correspondence of triangle meshes. *Int. J. Shape Model.* 13, 1, 101–124.
- M. Jin, J. Kim, F. Luo, and X. Gu. 2008. Discrete surface Ricci flow. *IEEE Trans. Visual. Comput. Graphics.* 14, 5, 1030–1043.
- A. Joshi, D. Shattuck, P. Thompson, and R. Leahy. 2007b. Surface-constrained volumetric brain registration using harmonic mappings. *IEEE Trans. Med. Imaging* 26, 12 (2007), 1657–1669.
- P. Joshi, M. Meyer, T. DeRose, B. Green, and T. Sanocki. 2007a. Harmonic coordinates for character articulation. In *SIGGRAPH'07*. 71–81.
- T. Ju, S. Schaefer, and J. D. Warren. 2005. Mean value coordinates for closed triangular meshes. *SIGGRAPH* 24, 3, 561–566.
- O. Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or. 2011. A survey on shape correspondence. *Comput. Graphics Forum* 30, 6, 1681–1707.
- M. Kass, A. Witkin, and D. Terzopoulos. 1988. Snakes: Active contour models. *Int. J. Computer Vision* 1, 4, 321–331.

- S. Katz and A. Tal. 2003. Hierarchical mesh decomposition using fuzzy clustering and cuts. In *ACM SIGGRAPH*. 954–961.
- M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. 2003. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Proceedings of the Symposium Geometry Processing (SGP'03)*. 156–164.
- L. Kharevych, B. Springborn, and P. Schröder. 2006. Discrete conformal mappings via circle patterns. *ACM Trans. Graphics* 25, 2, 412–438.
- A. Khodakovsky, N. Litke, and P. Schröder. 2003. Globally smooth parameterizations with low distortion. *ACM Trans. Graph.* 22, 3, 350–357.
- J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. Van Gool. 2010. Hough transform and 3D SURF for robust three dimensional classification. In *Proceedings of the European Conference on Computer Vision*. 589–602.
- L. Kobbelt, J. Vorsatz, U. Labsik, and H.-P. Seidel. 1999. A shrink wrapping approach to remeshing polygonal surfaces. *Comput. Graph. Forum* 18, 3, 119–130.
- V. Kraevoy, D. Julius, and A. Sheffer. 2007. Shuffler: Modeling with interchangeable parts. *Visual Comput.*
- V. Kraevoy and A. Sheffer. 2004. Cross-parameterization and compatible remeshing of 3D models. *ACM Trans. Graphics* 23, 3, 861–869.
- V. Kraevoy and A. Sheffer. 2005. Template-based mesh completion. In *Proceedings of the Symposium on Geometry Processing*. 13–22.
- V. Krayevoy and A. Sheffer. 2006. Variational, meaningful shape decomposition. In *ACM SIGGRAPH*. 50.
- V. Krayevoy, D. Julius, and A. Sheffer. 2007. Model composition from interchangeable components. In *Proceedings of the Pacific Conference on Computer Graphics and Applications*. 129–138.
- H. J. Kushner and G. G. Yin. 2003. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer-Verlag, New York.
- T.-H. Kwok, Y. Zhang, and C. C. L. Wang. 2012. Efficient optimization of common base domains for cross-parameterization. *IEEE Trans. Visual. Comp. Graphics* 18, 6, 914–924.
- Y.-K. Lai, S.-M. Hu, R. R. Martin, and P. L. Rosin. 2009. Rapid and effective segmentation of 3D models using random walks. *Comput. Aided Geom. Des.* 26, 6, 665–679.
- Y.-K. Lai, Q.-Y. Zhou, S.-M. Hu, and R. Martin. 2006. Feature sensitive mesh segmentation. In *Proceedings of the 2006 ACM Symposium on Solid and Physical Modeling*. 17–25.
- G. Lavoue, F. Dupont, and A. Baskurt. 2005. A new CAD mesh segmentation method, based on curvature tensor analysis. *Comput. Aided Des.* 37, 10, 975–987.
- F. Lazarus and A. Verroust. 1998. Three-dimensional metamorphosis: A survey. *Visual Comput.* 14, 8/9 (1998), 373–389.
- A. Lee, D. Dobkin, W. Sweldens, and P. Schröder. 1999. Multiresolution mesh morphing. In *SIGGRAPH*. 343–350.
- C. H. Lee, A. Varshney, and D. W. Jacobs. 2005b. Mesh saliency. *ACM Trans. Graph.* 24, 3, 659–666.
- T.-Y. Lee, C.-Y. Yao, H.-K. Chu, M.-J. Tai, and C.-C. Chen. 2006. Generating genus-n-to-m mesh morphing using spherical parameterization. *Comput. Animat. Virtual Worlds* 17, 3–4, 433–443.
- Y. Lee, H. Seok Kim, and S. Lee. 2002. Mesh parameterization with a virtual boundary. *Comput. Graphics* 26, 5, 677–686.
- Y. Lee, S. Lee, A. Shamir, D. Cohen-Or, and H.-P. Seidel. 2005a. Mesh scissoring with minima rule and part salience. *Comput. Aided Geom. Des.* 22, 5, 444–465.
- M. Leordeanu and M. Hebert. 2005. A spectral technique for correspondence problems using pairwise constraints. In *Proceedings of the International Conference on Computer Vision (ICCV'05)*, Vol. 2. 1482–1489.
- B. Lévy. 2003. Dual domain extrapolation. *ACM Trans. Graph.* 22, 3, 364–369.
- B. Lévy, S. Petitjean, N. Ray, and J. Maillot. 2002. Least squares conformal maps for automatic texture atlas generation. In *SIGGRAPH*. 362–371.
- B. Li, X. Li, K. Wang, and H. Qin. 2010. Generalized polycube trivariate splines. In *Proceedings of the 2010 Shape Modeling International Conference*. 261–265.
- B. Li, X. Li, K. Wang, and H. Qin. 2013. Surface mesh to volumetric spline conversion with generalized polycubes. *IEEE Trans. Visual. Comput. Graphics* 19, 9, 1539–1551.
- H. Li, B. Adams, L. Guibas, and M. Pauly. 2009. Robust single-view geometry and motion reconstruction. *ACM Trans. Graph.* 28, 5.
- H. Li, R. W. Sumner, and M. Pauly. 2008. Global correspondence optimization for non-rigid registration of depth scans. *Comput. Graphics Forum (Proc. SGP'08)* 27, 5, 1421–1430.
- X. Li. 2008. *A Shape Mapping Framework for Graphics and Visual Computing*. Ph.D. Dissertation. Stony Brook, NY. UMI Order Number: AAI3340114.

- X. Li, Yunfan Bao, X. Guo, Miao Jin, X. Gu, and H. Qin. 2008a. Globally optimal surface mapping for surfaces with arbitrary topology. *IEEE Trans. Visual. Comput. Graphics* 14, 4, 805–819.
- X. Li, X. Gu, and H. Qin. 2008b. Surface matching using consistent pants decomposition. In *Proceedings of the ACM Symposium on Solid and Physical Modeling*. 125–136.
- X. Li, X. Gu, and H. Qin. 2009. Surface mapping using consistent pants decomposition. *IEEE Trans. Visual. Comput. Graphics* 15, 4, 558–571.
- X. Li, X. Guo, H. Wang, Y. He, X. Gu, and H. Qin. 2007. Harmonic volumetric mapping for solid modeling applications. In *Proceedings of the ACM Symposium on Solid and Physical Modeling*. 109–120.
- X. Li and I. Guskov. 2005. Multi-scale features for approximate alignment of point-based surfaces. In *Proceedings of the Symposium on Geometry Processing*. 217–226.
- X. Li, Y. He, X. Gu, and H. Qin. 2006. Curves-on-surface: A general shape comparison framework. In *Proceedings of the IEEE International Conference of Shape Modeling*. 352–357.
- X. Li, T. W. Woon, T. S. Tan, and Z. Huang. 2001. Decomposing polygon meshes for interactive applications. In *Proceedings of the Symposium on Interactive 3D Graphics*. 35–42.
- X. Li, H. Xu, S. Wan, Z. Yin, and W. Yu. 2010. Feature-aligned harmonic volumetric mapping using MFS. *Comput. Graphics* 34, 3, 242–251.
- X. Li, Z. Yin, L. Wei, S. Wan, W. Yu, and M. Li. 2011. Symmetry and template guided completion of damaged skulls. *Comput. Graphics* 35, 4, 885–893.
- Y. Li, Y. Liu, W. Xu, W. Wang, and B. Guo. 2012. All-hex meshing using singularity-restricted field. *ACM Trans. Graphics* 31, 6, 177:1–177:11.
- J.-M. Lien and N. M. Amato. 2006. Approximate convex decomposition of polygons. *Comput. Geom.* 35, 1–2, 100–123.
- J.-M. Lien, J. Keyser, and N. M. Amato. 2006. Simultaneous shape decomposition and skeletonization. In *Proceedings of the ACM Symposium on Solid and Physical Modeling*. 219–228.
- T. Lindeberg. 1998. Feature detection with automatic scale selection. *Int. J. Comput. Vision* 30, 2, 79–116.
- Y. Lipman and T. Funkhouser. 2009. MoBius voting for surface correspondence. *ACM Trans. Graphics* 28, 3, 72:1–72:12.
- Y. Lipman, R. M. Rustamov, and T. A. Funkhouser. 2010. Biharmonic distance. *ACM Trans. Graphics* 29, 3, 27:1–27:11.
- R. Liu, V. Jain, and H. Zhang. 2006. Subsampling for efficient spectral mesh processing. In *CGI*. 172–184.
- R. Liu and H. Zhang. 2004. Segmentation of 3D meshes through spectral clustering. In *Proceedings of the Pacific Conference on Computer Graphics and Applications*. 298–305.
- R. Liu and H. Zhang. 2007. Mesh segmentation via spectral embedding and contour analysis. *Comput. Graphics Forum (Special Issue of Eurographics 2007)* 26, 3, 385–394.
- Y. Liu. 2007. A mean field annealing approach to accurate free form shape matching. *Pattern Recognit.* 40, 9, 2418–2436.
- E. M. Loiola, N. M. Maia de Abreu, P. O. Boaventura-Netto, P. Hahn, and T. Querido. 2007. A survey for the quadratic assignment problem. *Eur. J. Oper. Res.* 176, 2, 657–690.
- D. Lowe. 2004. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision* 60, 2, 91–110.
- L. M. Lui, K. C. Lam, S.-T. Yau, and X. Gu. 2012. Teichmüller extremal mapping and its applications to landmark matching registration. *arXiv:1211.256*.
- W.-C. Ma, F.-C. Wu, and M. Ouhyoung. 2003. Skeleton extraction of 3D objects with radial basis functions. In *Proceedings of Shape Modeling International*. 207–215.
- G. Malandain and S. Fernandez-Vidal. 1998. Euclidean skeletons. *Image Vision Comput.* 16, 5, 317–327.
- A. P. Mangan and R. T. Whitaker. 1999. Partitioning 3D surface meshes using watershed segmentation. *IEEE Trans. Visual. Comput. Graphics* 5, 4, 308–321.
- T. Martin, G. Chen, S. Musuvathy, E. Cohen, and C. D. Hansen. 2012. Generalized swept mid-structure for polygonal models. *Comput. Graphics Forum* 31, 2, 805–814.
- T. Martin, E. Cohen, and R. M. Kirby. 2008. Volumetric parameterization and trivariate b-spline fitting using harmonic functions. In *Proceedings of the ACM Solid and Physical Modeling*. 269–280.
- D. Mateus, R. Horaud, D. Knossow, F. Cuzzolin, and E. Boyer. 2008. Articulated shape matching using Laplacian eigenfunctions and unsupervised point registration. In *CVPR*. 1–8.
- J. C. McBride and B. B. Kimia. 2003. Archaeological fragment reconstruction using curve-matching. In *Proceedings of the Computer Vision and Pattern Recognition Workshop*, Vol. 1. 3–3.
- K. K. Meenakshi, S. Srihari, and A. Xu. 2004. Offline signature verification and identification using distance statistics. *Int. J. Pattern Recognit. Artif. Intell.* 18, 7, 1339–1360.

- C. T. Metz, S. Klein, M. Schaap, T. van Walsum, and W. J. Niessen. 2011. Nonrigid registration of dynamic medical imaging data using nD + t B-splines and a groupwise optimization approach. *Med. Image Anal.* 15, 2, 238–249.
- A. Mian, M. Bennamoun, and R. Owens. 2010. On the repeatability and quality of keypoints for local feature-based 3D object retrieval from cluttered scenes. *Int. J. Comput. Vision* 89, 2–3, 348–361.
- K. Mikolajczyk and C. Schmid. 2005. A performance evaluation of local descriptors. *IEEE Trans. PAMI* 27, 10, 1615–1630.
- J. W. Milnor. 1963. *Morse Theory*. Princeton University Press, Princeton, NJ.
- J. Mitani and H. Suzuki. 2004. Making papercraft toys from meshes using strip-based approximate unfolding. *ACM Trans. Graphics* 23, 3, 259–263.
- J. J. Moré and D. J. Thuente. 1994. Line search algorithms with guaranteed sufficient decrease. *ACM Trans. Math. Softw.* 20, 3, 286–307.
- J. Munkres. 1957. Algorithms for the assignment and transportation problems. *J. Soc. Indust. Appl. Math.* 5, 1, 32–38.
- M. Nieser, U. Reitebuch, and K. Polthier. 2011. CubeCover—Parameterization of 3D volumes. *Comput. Graphics Forum* 30, 5, 1397–1406.
- J. Novatnack and K. Nishino. 2008. Scale-dependent/invariant local 3D shape descriptors for fully automatic registration of multiple sets of range images. In *ECCV (3)*. 440–453.
- M. Novotni and R. Klein. 2003. 3D Zernike descriptors for content based shape retrieval. In *Proceedings of the ACM Symposium on Solid Modeling and Applications*. ACM, New York, NY, 216–225.
- S. Osher and R. Fedkiw. 2002. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, New York, NY.
- M. Ovsjanikov, J. Sun, and L. Guibas. 2008. Global intrinsic symmetries of shapes. In *Proceedings of the Symposium on Geometry Processing*. 1341–1348.
- T. Pajdla and L. V. Gool. 1995. Matching of 3-D curves using semi-differential invariants. In *Proceedings of the 5th International Conference on Computer Vision*. IEEE Computer Society Press, Los Alamitos, CA, 390–395.
- V. Pascucci, G. Scorzelli, P.-T. Bremer, and A. Mascarenhas. 2007. Robust on-line computation of Reeb graphs: Simplicity and speed. *ACM Trans. Graphics* 26, 3, 58.1–58.9.
- G. Patane, M. Spagnuolo, and B. Falcidieno. 2008. Reeb graph computation based on a minimal contouring. In *Shape Modeling International*. 73–82.
- M. Pauly, N. J. Mitra, J. Giesen, M. Gross, and L. Guibas. 2005. Example-based 3D scan completion. In *Symposium on Geometry Processing*. 23–32.
- Y. Pekelny and C. Gotsman. 2008. Articulated object reconstruction and markerless motion capture from depth video. *Comput. Graphics Forum* 27, 2, 399–408.
- U. Pinkall and K. Polthier. 1993. Computing discrete minimal surfaces and their conjugates. In *Experimental Mathematics*, Vol. 2. 15–36.
- J. P. W. Pluim and J. M. Fitzpatrick. 2003. Image registration. *IEEE Trans. Med. Imaging* 22, 11, 1341–1343.
- J. Podolak, P. Shilane, A. Golovinskiy, S. Rusinkiewicz, and T. Funkhouser. 2006. A planar-reflective symmetry transform for 3D shapes. In *ACM SIGGRAPH*. 549–559.
- E. Praun and H. Hoppe. 2003. Spherical parameterization and remeshing. In *SIGGRAPH '03*. 340–349.
- E. Praun, W. Sweldens, and P. Schröder. 2001. Consistent mesh parameterizations. In *SIGGRAPH*. 179–184.
- D. Quynh, Y. He, S.-Q. Xin, and Z. Chen. 2012. An intrinsic algorithm for computing geodesic distance fields on triangle meshes with holes. *Graphical Models* 74, 4, 209–220.
- R. Raab, C. Gotsman, and A. Sheffer. 2004. Virtual woodwork: Generating bead figures from 3D models. *Int. J. Shape Model.* 10, 1, 1–30.
- N. Ray, W. Li, B. Lévy, A. Sheffer, and P. Alliez. 2006. Periodic global parameterization. *ACM Trans. Graphics* 25, 4, 1460–1485.
- D. Reniers and A. Telea. 2007. Skeleton-based hierarchical shape segmentation. In *Proceedings of the IEEE International Conference on Shape Modeling and Applications*. 179–188.
- M. Reuter, F.-E. Wolter, and N. Peinecke. 2005. Laplace-spectra as fingerprints for shape matching. In *Proceedings of the Symposium on Solid and Physical Modeling*. 101–106.
- L. Reyes, G. Medioni, and E. Bayro. 2007. Registration of 3D points using geometric algebra and tensor voting. *Int. J. Comput. Vision* 75, 3, 351–369.
- S. Ruiz-Correa, L. G. Shapiro, and M. Melia. 2001. A new signature-based method for efficient 3-D object recognition. In *Proceedings of CVPR*, Vol. 1. 769–776.
- S. Rusinkiewicz and M. Levoy. 2001. Efficient variants of the ICP algorithm. In *Proceedings of the 3rd International Conference on 3D Digital Imaging and Modeling*. 145–152.

- R. M. Rustamov. 2007. Laplace-Beltrami eigenfunctions for deformation invariant shape representation. In *Proceedings of the Symposium on Geometry Processing*. 225–233.
- S. Saba, I. Yavneh, C. Gotsman, and A. Sheffer. 2005. Practical spherical embedding of manifold triangle meshes. In *Proceedings of the International Conference on Shape Modeling and Applications*. 258–267.
- Y. Sahillioglu. 2012. *Algorithms for 3D Isometric Shape Correspondence*. Ph.D. dissertation. Koç Üniversitesi Graduate School of Sciences & Engineering, Istanbul, Turkey.
- Y. Sahillioglu and Y. Yemez. 2010. 3D Shape correspondence by isometry-driven greedy optimization. In *CVPR*. 453–458.
- P. V. Sander, J. Snyder, S. J. Gortler, and H. Hoppe. 2001. Texture mapping progressive meshes. In *SIGGRAPH'01*. 409–416.
- F. R. Schmidt, M. Clausen, and D. Cremers. 2006. Shape matching by variational computation of geodesics on a manifold. In *Proceedings of the 28th Conference on Pattern Recognition (DAGM'06)*. 142–151.
- R. Schoen and S. T. Yau. 1997. *Lectures on Harmonic Maps*. International Press, Somerville, MA.
- J. Schreiner, A. Asirvatham, E. Praun, and H. Hoppe. 2004. Inter-surface mapping. *SIGGRAPH*. 23, 3, 870–877.
- T. B. Sebastian, P. N. Klein, and B. B. Kimia. 2003. On aligning curves. *IEEE Trans. Pattern Anal. Mach. Intell.* 25, 1, 116–125.
- T. Sederberg and S. Parry. 1986. Free-form deformation of solid geometric models. *SIGGRAPH* 20, 4, 151–160.
- S. Shalom, L. Shapira, A. Shamir, and D. Cohen-Or. 2008. Part analogies in sets of objects. In *Proceedings of the Symposium on 3D Object Retrieval*. 33–40.
- A. Shamir. 2008. A survey on Mesh Segmentation Techniques. *Comput. Graphics Forum* 27, 6, 1539–1556.
- L. Shapira, A. Shamir, and D. Cohen-Or. 2008. Consistent mesh partitioning and skeletonisation using the shape diameter function. *Vis. Comput.* 24, 4, 249–259.
- A. Shapiro and A. Tal. 1998. Polyhedron realization for shape transformation. *Visual Comput.* 14, 8, 429–444.
- A. Sheffer, C. Gotsman, and N. Dyn. 2004. Robust spherical parameterization of triangular meshes. *Computing* 72, 1–2, 185–193.
- A. Sheffer, B. Lévy, M. Mogilnitsky, and A. Bogomyakov. 2005. ABF++: Fast and robust angle based flattening. *ACM Trans. Graph.* 24, 2, 311–330.
- A. Sheffer, E. Praun, and K. Rose. 2006. Mesh parameterization methods and their applications. *Found. Trends. Comput. Graph. Vis.* 2, 2, 105–171.
- A. Sheffer and E. De Sturler. 2000. Surface parameterization for meshing by triangulation flattening. In *Proceedings of the 9th International Meshing Roundtable*. 161–172.
- S. Shlafman, A. Tal, and S. Katz. 2002. Metamorphosis of polyhedral surfaces using decomposition. In *Comput. Graphics Forum*. 219–228.
- K. Siddiqi and S. Pizer. 2009. *Medial Representations: Mathematics, Algorithms and Applications*. Springer, New York, NY.
- I. Sipiran and B. Bustos. 2010. A robust 3D interest points detector based on Harris operator. In *Proceedings of 3D Object Retrieval*. 7–14.
- F. Stein and G. Medioni. 1992. Structural indexing: Efficient 3-D object recognition. *IEEE Trans. PAMI* 14, 125–145.
- K. Stephenson. 2005. *Introduction to circle packing*. Cambridge University Press, New York, NY.
- R. W. Sumner and J. Popović. 2004. Deformation transfer for triangle meshes. In *ACM Trans. Graph. SIGGRAPH'04*. 399–405.
- J. Sun, M. Ovsjanikov, and L. Guibas. 2009. A concise and provably informative multi-scale signature based on heat diffusion. In *Proceedings of the Symposium on Geometry Processing*. 1383–1392.
- H. Sundar, D. Silver, N. Gagvani, and S. Dickinson. 2003. Skeleton based shape matching and retrieval. In *Shape Modeling International*. 130–139.
- S. Svensson, I. Nystrom, and G. Sanniti di Baja. 2002. Curve skeletonization of surface-like objects in 3D images guided by voxel classification. *Pattern Recognit. Lett.* 23, 12, 1419–1426.
- G. Tam, Z. Cheng, Y. Lai, F. Langbein, Y. Liu, D. Marshall, R. Martin, X. Sun, and P. Rosin. 2013. Registration of 3D point clouds and meshes: A survey from rigid to nonrigid. *IEEE Trans. Vis. Comput. Graphics* 19, 7, 1199–1217.
- A. Tevs, A. Berner, M. Wand, I. Ihrke, and H.-P. Seidel. 2011. Intrinsic shape matching by planned landmark sampling. *Comput. Graphics Forum* 30, 2, 543–552.
- A. Tevs, M. Bokeloh, M. Wand, A. Schilling, and H.-P. Seidel. 2009. Isometric registration of ambiguous and partial data. In *Proceedings of CVPR*. 1185–1192.

- J. Tierny, J.-P. Vandeborre, and M. Daoudi. 2007. Topology driven 3D mesh hierarchical segmentation. In *Shape Modeling and International*. 215–220.
- F. Tombari, S. Salti, and L. Di Stefano. 2013. Performance evaluation of 3D keypoint detectors. *Intl. J. Comput. Vision* 102, 1–3, 198–220.
- Y. Tong, P. Alliez, D. Cohen-Steiner, and M. Desbrun. 2006. Designing quadrangulations with discrete harmonic forms. In *Proceedings of the Symposium on Geometry Processing*. 201–210.
- T. Tung and T. Matsuyama. 2010. Dynamic surface matching by geodesic mapping for 3D animation transfer. In *CVPR*. 1402–1409.
- T. Tung and F. Schmitt. 2005. The augmented multiresolution Reeb graph approach for content-based retrieval of 3D shapes. *Int. J. Shape Model.* 11, 1, 91–120.
- W. Tutte. 1963. How to draw a graph. In *Proceedings of the London Mathematical Society*. 743–767.
- R. Unnikrishnan and M. Hebert. 2008. Multi-scale interest regions from unorganized point clouds. In *Proceedings of the Workshop on Search in 3D (S3D'08)*. 1–8.
- B. Vallet and B. Lévy. 2008. Spectral geometry processing with manifold harmonics. *Comput. Graphics Forum* 27, 2, 251–260.
- T. Várady, M. Facello, and Z. Terék. 2007. Automatic extraction of surface structures in digital shape reconstruction. *Comput. Aided Des.* 39, 5, 379–388.
- M. Vieira and K. Shimada. 2005. Surface mesh segmentation and smooth surface extraction through region growing. *Comput. Aided Geom. Des.* 22, 8, 771–792.
- D. Vlastic, I. Baran, W. Matusik, and J. Popović. 2008. Articulated mesh animation from multi-view silhouettes. *ACM Trans. Graphics* 27, 3, 97:1–97:9.
- M. Walker, L. Shao, and R. Volz. 1991. Estimating 3-D location parameters using dual number quaternions. *CVGIP: Image Underst.* 54, 3, 358–367.
- S. Wan, T. Ye, M. Li, H. Zhang, and X. Li. 2012. Efficient spherical parameterization using progressive optimization. In *Computational Visual Media. Lecture Notes in Computer Science*, Vol. 7633. 170–177.
- S. Wan, T. Ye, M. Li, H. Zhang, and X. Li. 2013. An efficient spherical mapping algorithm and its application on spherical harmonics. *SCIENCE CHINA: Information Sciences* 56, 9, 1–10.
- S. Wan, Z. Yin, K. Zhang, H. Zhang, and X. Li. 2011. A topology-preserving optimization algorithm for polycube mapping. *Comput. Graphics* 35, 3, 639–649.
- M. Wand, B. Adams, M. Ovsjanikov, A. Berner, M. Bokeloh, P. Jenke, L. Guibas, H.-P. Seidel, and A. Schilling. 2009. Efficient reconstruction of nonrigid shape and motion from real-time 3D scanner data. *ACM Trans. Graph.* 28, 2, 15:1–15:15.
- M. Wand, P. Jenke, Q. Huang, M. Bokeloh, L. Guibas, and A. Schilling. 2007. Reconstruction of deforming geometry from time-varying point clouds. In *Proceedings of the Symposium on Geometry Processing*. 49–58.
- H. Wang, Ying He, Xin Li, X. Gu, and H. Qin. 2008b. Polycube splines. *Comput. Aided Des.* 40, 6, 721–733.
- H. Wang, Ying He, X. Li, X. Gu, and H. Qin. 2009. Geometry-aware domain decomposition for T-spline-based manifold modeling. *Comput. Graphics* 33, 3, 359–368.
- K. Wang, X. Li, B. Li, H. Xu, and H. Qin. 2012. Restricted trivariate polycube splines for volumetric data modeling. *IEEE Trans. Visual. Comput. Graphics* 18, 5, 703–716.
- Y. Wang, X. Gu, T. F. Chan, P. M. Thompson, and S. T. Yau. 2004. Volumetric harmonic brain mapping. In *Proceedings of the IEEE International Symposium on Biomedical Imaging: Macro to Nano*. 1275–1278.
- Y. Wang, M. Gupta, S. Zhang, S. Wang, X. Gu, D. Samaras, and P. Huang. 2008a. High resolution tracking of non-rigid motion of densely sampled 3D data using harmonic maps. *Int. J. Comput. Vision* 76, 3, 283–300.
- O. Weber, A. Myles, and D. Zorin. 2012. Computing extremal quasiconformal maps. *Comp. Graphics Forum* 31, 5, 1679–1689.
- K. Wu and M. D. Levine. 1996. 3D part segmentation using simulated electrical charge distributions. *ICPR* 1, 14.
- K. Wu and M. D. Levine. 1997. 3D part segmentation using simulated electrical charge distributions. *IEEE Trans. Pattern Anal. Mach. Intell.* 19, 1223–1235.
- J. Xia, Y. He, X. Yin, S. Han, and X. Gu. 2010. Direct-product volumetric parameterization of handlebodies via harmonic fields. In *Proceedings of the International Conference on Shape Modeling and Applications*. 3–12.
- H. Xu, P. Chen, W. Yu, A. Sawant, S. Iyengar, and X. Li. 2012. Feature-aligned 4D spatiotemporal image registration. In *Proceedings of the International Conference on Pattern Recognition (ICPR '12)*. 2639–2642.

- H. Xu and X. Li. 2013a. Consistent feature-aligned 4D image registration for respiratory motion modeling. In *Proceedings of the IEEE International Symposium on Biomedical Imaging (ISBI '13)*. 580–583.
- H. Xu and X. Li. 2013b. A symmetric 4D registration algorithm for respiratory motion modeling. In *Proceedings of the International Conference Medical Image Computing and Computer Assisted Intervention (MICCAI'13)*. 149–155.
- H. Xu, W. Yu, S. Gu, and X. Li. 2013a. Biharmonic volumetric mapping using fundamental solutions. *IEEE Transactions on Visualization and Computer Graphics* 19, 5, 787–798.
- H. Xu, H. Zhang, and X. Li. 2013b. *Adaptive Stochastic Conjugate Gradient Optimization for Temporal Medical Image Registration*. Technical Report, Louisiana State University, Baton Rouge, LA.
- H. Yamauchi, S. Gumhold, R. Zayer, and H.-P. Seidel. 2005a. Mesh segmentation driven by gaussian curvature. *Visual Computer* 21, 8–10, 649–658.
- H. Yamauchi, S. Lee, Y. Lee, Y. Ohtake, A. Belyaev, and H.-P. Seidel. 2005b. Feature sensitive mesh segmentation with mean shift. In *Proceedings of the International Conference on Shape Modeling and Applications*. 238–245.
- Z. Yin, L. Wei, M. Manhein, and X. Li. 2011. An automatic assembly and completion framework for fragmented skulls. In *Proceedings of the International Conference on Computer Vision (ICCV'11)*. 2532–2539.
- X. Ying, X. Wang, and Y. He. 2013. Saddle vertex graph (SVG): A novel solution to the discrete geodesic problem. *ACM Trans. Graph. (SIGGRAPH ASIA'13)* 32, 6, 170.
- W. Yu and X. Li. 2011. Computing 3D shape guarding and star decomposition. *Comput. Graph. Forums* 30, 7, 2087–2096.
- W. Yu and X. Li. 2014. A new polycube shape optimization algorithm for hex mesh generation. In *Proceedings of the Workshop of Structured-Meshing: Theory, Applications, and Evaluation, International Conference on Computer Animation and Social Agents (CASA'14)*.
- W. Yu, T. Ye, M. Li, and X. Li. 2010. Spherical harmonic decomposition for surfaces of arbitrary topology. In *Proceedings of the International Conference on Computer Science and Education*. 215–220.
- W. Yu, K. Zhang, S. Wan, and X. Li. 2013. Optimizing polycube domain construction for hexahedral remeshing. *Comput. -Aided Des.* 46, 58–68.
- A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud. 2009. Surface feature detection and description with applications to mesh matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*. 373–380.
- R. Zayer, C. Rossli, and H.-P. Seidel. 2006. Curvilinear spherical parameterization. In *Proceedings of the International Conference on Shape Modeling and Applications*. 57–64.
- W. Zeng, X. Li, S.-T. Yau, and X. Gu. 2007. Conformal spherical parameterization for high genus surfaces. *Comm. Inf. Syst.* 7, 3, 273–286.
- E. Zhang, K. Mischaikow, and G. Turk. 2005. Feature-based surface parameterization and texture mapping. *ACM Trans. Graphics* 24, 1, 1–27.
- H. Zhang, O. Kaick, and R. Dyer. 2010. Spectral mesh processing. *Comput. Graphics Forum* 29, 6, 1865–1894.
- H. Zhang and R. Liu. 2005. Mesh segmentation via recursive and visually salient spectral cuts. In *Proceedings of Vision, Modeling, and Visualization*. 429–436.
- K. Zhang and X. Li. 2012. Optimizing geometry-aware pants decomposition. In *Proceedings of the Pacific Conference on Computer Graphics and Applications*. 11–16.
- K. Zhang and X. Li. 2014. A graph-based optimization algorithm for fragmented image reassembly. *Graphical Models* 76, 5, 484–495.
- M. Zhang, F. Li, X. Wang, Z. Wu, S.-Q. Xin, L.-M. Lui, L. Shi, D. Wang, and Y. He. 2014. Automatic registration of vestibular systems with exact landmark correspondence. *Graphical Models* 76, 5, 532–541.
- Z. Zhang. 1994. Iterative point matching for registration of free-form curves and surfaces. *Int. J. Comput. Vision* 13, 2, 119–152.
- X. Zhao, Z. Su, X. Gu, A. Kaufman, J. Sun, J. Gao, and F. Luo. 2013. Area-preservation mapping using optimal mass transport. *IEEE Trans. Vis. Comput. Graphics* 19, 12, 2838–2847.
- M. Zhong, T. Hou, and H. Qin. 2012. A hierarchical approach to high-quality partial shape registration. In *Proceedings of the 2012 21st International Conference on Pattern Recognition (ICPR'12)*. 113–116.
- Y. Zhong. 2009. Intrinsic shape signatures: A shape descriptor for 3D object recognition. In *Proceedings of the ICCV Workshops*. 689–696.
- Y. Zhong and Z. Huang. 2004. Decomposing polygon meshes by means of critical points. In *Proceedings of the 10th International Multimedia Modelling Conference*. 187–195.
- M. Zöckler, D. Stalling, and H.-C. Hege. 2000. Fast and intuitive generation of geometric shape transitions. *Vis. Comput.* 16, 5, 241–253.

- D. Zorin, P. Schroder, T. Derose, L. Kobbelt, A. Levin, and W. Sweldens. 2000. Subdivision for modeling and animation. In *ACM Siggraph Course*.
- E. Zuckerberger, A. Tal, and S. Shlafman. 2002. Polyhedral surface decomposition with applications. *Comput. Graphics* 26, 5, 733–743.

Received January 2014; revised June 2014; accepted September 2014