Chapter 5
Feature-based Methods in 3D Shape Analysis

Alexander M. Bronstein, Michael M. Bronstein, and Maks Ovsjanikov

Abstract The computer vision and pattern recognition communities have recently witnessed a surge in feature-based methods for numerous applications including object recognition and image retrieval. Similar concepts and analogous approaches are penetrating the world of 3D shape analysis in a variety of areas including non-rigid shape retrieval and matching. In this chapter, we present both mature concepts and the state-of-the-art of feature-based approaches in 3D shape analysis. In particular, approaches to the detection of interest points and the generation of local shape descriptors is discussed. A wide range of methods is covered including those based on curvature, those based on difference-of-Gaussian scale space, and those that employ recent advances in heat kernel methods.

5.1 Introduction

In computer vision and pattern recognition jargon, the term *features* is often used to refer to persistent elements of a 2D image (such as corners or sharp edges), which capture most of the relevant information and allow one to perform object analysis. In the last decade, feature-based methods (such as the scale invariant feature transform (SIFT) [51] and similar algorithms [55, 4]) have become a standard and broadly-used paradigm in various applications, including retrieval and matching (e.g. for multiview geometry reconstruction), due to their relative simplicity, flexibility, and excellent performance in practice.

Alexander M. Bronstein · Michael M. Bronstein
Department of Computer Science, Technion - Israel Institute of Technology, Haifa 32000, Israel. e-mail: bron@cs.technion.ac.il, mbron@cs.technion.ac.il

Maks Ovsjanikov
Department of Computer Science, Stanford University, USA. e-mail: maks@stanford.edu

A similar trend is emerging in 3D shape analysis in a variety of areas including non-rigid shape retrieval and shape matching. While in some cases computer vision methods are straightforwardly applicable to 3D shapes [50, 45], in general, some fundamental differences between 2D and 3D shapes require new and different methods for shape analysis.

One of the distinguishing characteristics that make computer vision techniques that work successfully in 2D image analysis not straightforwardly applicable in 3D shape analysis is the difference in shape representations. In computer vision, it is common to work with a 2D image of a physical object, representing both its geometric and photometric properties. Such a representation simplifies the task of shape analysis by reducing it to simple image processing operations, at the cost of losing information about the object's 3D structure, which cannot be unambiguously captured in a 2D image. In computer graphics and geometry processing, it is assumed that the 3D geometry of the object is explicitly given. Depending on application, the geometric representation of the object can differ significantly. For example, in graphics it is common to work with triangular meshes or point clouds; in medical applications with volumes and implicit representations.

Furthermore, 3D shapes are usually poorer in high-frequency information (such as edges in images), and being generally non-Euclidean spaces, many concepts natural in images (edges, directions, etc.), do not straightforwardly generalize to shapes.

Most feature-based approaches can be logically divided into two main stages: location of stable, repeatable points that capture most of the relevant shape information (*feature detection*[1]) and representation of the shape properties at these points (*feature description*). Both processes depend greatly on shape representation as well as on the application at hand.

In 2D image analysis, the typical use of features is to describe an object independently of the way it is seen by a camera. Features found in images are geometric discontinuities in the captured object (edges and corners) or its photometric properties (texture). Since the difference in viewpoint can be *locally* approximated as an affine transformation, feature detectors and descriptors in images are usually made *affine invariant*.

In 3D shape analysis, features are typically based on geometry rather than appearance. The problems of shape correspondence and similarity require the features to be stable under natural transformations that an object can undergo, which may include not only changes in pose, but also non-rigid bending. If the deformation is inelastic, it is often referred to as *isometric* (distance-preserving), and feature-based methods coping with such transformations as *isometry-invariant*; if the bending also involves connectivity changes, the feature detection and description algorithms are called *topology-invariant*.

[1] In some literature, this is also known as *interest point* detection or *keypoint detection*.

The main challenge of feature-based 3D shape analysis can be summarized as finding a set of features that can be found repeatably on shapes undergoing a wide class of transformations on the one hand and carry sufficient information to allow using these features to find correspondence and similarity (among other tasks) on the other.

5.1.1 Applications

Two archetypal problems in shape analysis addressed by feature-based methods are *shape similarity* and *correspondence*. The former underlies many pattern recognition applications, where we have to distinguish between different geometric objects (e.g. in 3D face recognition [13]). A particularly challenging setting of the shape similarity problem appears in *content-based shape retrieval*, an application driven by the availability of large public-domain databases of 3D models, such as *Google 3D Warehouse*, which have created the demand for shape search and retrieval algorithms capable of finding similar shapes in the same way a search engine responds to text queries (detailed discussion of this application appears in Chap. 7).

One of the notable advantages of feature-based approaches in shape retrieval is the possibility of representing a shape as a collection of primitive elements ("geometric words"), and using the well-developed methods from text search such as the *bag of features* (BOF) (or *bag of words*) paradigm [75, 23]. Such approaches are widely used in image retrieval and have been introduced more recently to shape analysis [17, 83]. The construction of a bag of features is usually performed in a few steps, depicted in Fig. 5.1. Firstly, the shape is represented as a collection of local feature descriptors (either dense or computed as a set of stable points following an optional stage of feature detection). Secondly, the descriptors are represented by *geometric words* from a *geometric vocabulary* using vector quantization. The geometric vocabulary is a set of representative descriptors, precomputed in advance. This way, each descriptor is replaced by the index of the closest geometric word in the vocabulary. Computing the histogram of the frequency of occurrence of geometric words gives the bag of features. Alternatively, a two-dimensional histogram of co-occurrences of pairs of geometric words (*geometric expressions*) can be used [17]. Shape similarity is computed as a distance between the corresponding bags of features. The bag of features representation is usually compact, easy to store and compare, which makes such approaches suitable for large-scale shape retrieval. Evaluation of shape retrieval performance (e.g. the robust large-scale retrieval benchmark [11] from the Shape Retrieval Contest (SHREC)) tests the robustness of retrieval algorithms on a large set of shapes with different simulated transformations, including non-rigid deformations.

Fig. 5.1 Construction of bags of features for feature-based shape retrieval.

Another fundamental problem in shape analysis is that of *correspondence* consisting of finding relations between similar points on two or more shapes. Finding correspondence between two shapes that would be invariant to a wide variety of transformations is usually referred to as *invariant shape correspondence*. Correspondence problems are often encountered in shape synthesis applications such as morphing. In order to morph one shape into the other, one needs to know which point on the first shape will be transformed into a point on the second shape, in other words, establishing a correspondence between the shapes. A related problem is *registration*, where the deformation bringing one shape into the other is explicitly sought for.

Feature-based methods for shape correspondence are based on first detecting features on two shapes between which correspondence is sought, and then match them by comparing the corresponding descriptors. The feature-based correspondence problem can be formulated as finding a map that maximizes the similarity between corresponding descriptors. The caveat of such an approach is that it may produce inconsistent matches, especially in shapes with repeating structure or symmetry: for example, points on the right and left sides of a human body can be swapped due to bilateral symmetry. A way to cope with this problem is to add some *global structure*, for example, pairwise geodesic or diffusion distance preservation constraint. Thus, this type of *minimum-distortion correspondence* tries to match simultaneously local structures (descriptors) and global structures (metrics), and can be found by an extension of the generalized multidimensional scaling (GMDS) algorithm [14, 82] or graph labeling [78, 84, 85]. Evaluation of correspondence finding algorithms typically simulates a one-to-one shape matching scenario, in which one of the shapes undergoes multiple modifications and transformations, and the quality of the correspondence is evaluated as the distance on the shape between the found matches and the known groundtruth correspondence. Notable benchmarks are the SHREC robust correspondence benchmark [12] and the Princeton correspondence benchmark [41].

5.1.2 Chapter Outline

In this chapter, we present an overview of feature-based methods in 3D shape analysis and their applications, classical as well as most recent approaches. The main emphasis is on heat-kernel based detection and description algorithms, a relatively recent set of methods based on a common mathematical model and falling under the umbrella of *diffusion geometry*. Detailed description, examples, figures, and problems in this chapter allows the implementation of these methods.

The next section outlines some prerequisite mathematical background, describing our notation and a number of important concepts in differential and diffusion geometry. Then the two main sections are presented: Sect. 5.3 discusses feature detectors, while Sect. 5.4 describes feature descriptors. The final sections give concluding remarks, research challenges and suggested further reading.

5.2 Mathematical Background

Throughout this chapter, an object is some subset of the ambient Euclidean space, $\Omega \subset \mathbb{R}^3$. In many cases (e.g. data acquired by a range scanner), we can access only the *boundary* $\partial\Omega$ of the object, which can be modeled as a two-dimensional smooth *manifold* or *surface*, denoted here by $\mathbb{X}$. Photometric information is given as a scalar or a vector field $\alpha : \mathbb{X} \to \mathbb{R}^d$ on the manifold and referred to as *texture*. If the surface is sampled at some discrete set of points $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subset \mathbb{X}$, then this representation is called a *point cloud*; if, in addition, connectivity information is available in the form of a simplicial complex (*triangulation*, consisting of a set of *edges* $(\mathbf{x}_i, \mathbf{x}_j) \in \mathbb{E}$ and *faces* $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) \in \mathbb{F}$), such a representation is called a *mesh*.

In medical applications, such as tomographic data analysis, information about the internal structure of the object in addition to its boundary is often available. A common representation in such applications is a *volumetric image*, which can be represented as a 3D matrix, where each voxel (3D pixel) describes the properties of the object (e.g. its penetrability by X-ray radiation). Segmentation algorithms applied to volumetric data used in medical applications often extract boundaries of 3D objects in *implicit form*, represented as level-sets of some function $f : \mathbb{R}^3 \to \mathbb{R}$.

5.2.1 Differential Geometry

Both the two-dimensional boundary surface and the three-dimensional volume enclosed by it can be modeled as, respectively, two- and three-dimensional

complete Riemannian sub-manifolds of $\mathbb{R}^3$. Every point $\mathbf{x}$ on the manifold $\mathbb{X}$ is assigned a *tangent space* $T_x\mathbb{X}$. For two dimensional surfaces, the vector $\mathbf{N}$ orthogonal to $T_x\mathbb{X}$ is called the *normal* to the surface at $\mathbf{x}$. The tangent space at each point is associated with a smooth inner product $g_x : T_x\mathbb{X} \times T_x\mathbb{X} \to \mathbb{R}$, usually referred to as the *metric tensor*. Denoting by $\mathbf{x} : \mathbb{U} \subseteq \mathbb{R}^2 \to \mathbb{R}^3$ the regular map embedding $\mathbb{X}$ into $\mathbb{R}^3$, the metric tensor can be expressed in coordinates as

$$g_{ij} = \frac{\partial \mathbf{x}^{\mathrm{T}}}{\partial u_i} \frac{\partial \mathbf{x}}{\partial u_j}, \tag{5.1}$$

where u_i are the coordinates of U. The metric tensor relates infinitesimal displacements du in the parametrization domain $\mathbb{U}$ to displacement on the manifold,

$$dp^2 = g_{11} {du_1}^2 + 2g_{12} du_1 du_2 + g_{22} {du_2}^2. \tag{5.2}$$

This quadratic form is usually referred to as the *first fundamental form* and it provides a way to define length structures on the manifold. Given a curve $C : [0, T] \to X$, its length can be expressed as

$$L(C) = \int_0^T g(\dot{C}(t), \dot{C}(t))^{1/2}_{C(t)} dt, \tag{5.3}$$

where $\dot{C}$ denotes the velocity vector. *Minimal geodesics* are the minimizers of $L(C)$, giving rise to the *geodesic distances*

$$d(\mathbf{x}, \mathbf{x}') = \min_{C \in \Gamma(\mathbf{x}, \mathbf{x}')} L(C), \tag{5.4}$$

where $\Gamma(\mathbf{x}, \mathbf{x}')$ is the set of all admissible paths between the points $\mathbf{x}$ and $\mathbf{x}'$ on the surface $\mathbb{X}$ (due to completeness assumption, the minimizer always exists). Structures expressible solely in terms of the metric tensor g are called *intrinsic*. For example, the geodesic can be expressed in this way. The importance of intrinsic structures stems from the fact that they are invariant under isometric transformations (bendings) of the shape. In an isometrically bent shape, the geodesic distances are preserved, which is a property that allows the design of isometrically invariant shape descriptors [31].

The metric tensor also allows the definition of differential operations on the tangent space. Given a smooth scalar field $f : \mathbb{X} \to \mathbb{R}$, its *(intrinsic) gradient* $\nabla_X f$ at point $\mathbf{x}$ is defined through the relation $f(\mathbf{x} + \mathbf{dv}) = f(\mathbf{x}) + g_x(\nabla_X f(\mathbf{x}), \mathbf{dv})$, where $\mathbf{dv} \in T^*_x\mathbb{X}$ is an infinitesimal tangent vector. For a given tangent vector $\mathbf{v}$, the quantity

$$D_v f = \lim_{\epsilon \to 0} \frac{f(\mathbf{x} + \epsilon \mathbf{v}) - f(\mathbf{x})}{\epsilon \sqrt{g_x(\mathbf{v}, \mathbf{v})}} \tag{5.5}$$

is called the *directional derivative* of f at point $\mathbf{x}$ in the direction $\mathbf{v}$.

5.2.2 Curvature of Two-dimensional Surfaces

Given a curve $\gamma : [0, L] \to \mathbb{R}^3$, its first-order and second-order derivatives with respect to the parameter, $\dot{\gamma}$ and $\ddot{\gamma}$, are called the *tangent* and *curvature* vectors, respectively. The magnitude of $\ddot{\gamma}(t)$ measures the curvature of γ at a point. The curvature of a surface at a point $\mathbf{x}$ can be expressed in terms of curves passing through it confined to the surface. Every direction $\mathbf{v} \in T_x\mathbb{X}$ can be associated with a curve γ such that $\gamma(0) = \mathbf{x}$ and $\dot{\gamma}(0) = \mathbf{v}$, and, thus, with a curvature vector $\ddot{\gamma}(0)$. The projection of the curvature vector on the tangent plane is called the *geodesic curvature*, and it vanishes if and only if γ is a geodesic. The projection $\kappa_n = \mathsf{P}_N\ddot{\gamma}(0)$ of the curvature vector on the normal is called the *normal curvature*. The minimum and the maximum values of κ_n are called the *principal curvatures* $\kappa_1 \leq \kappa_2$, and the corresponding directions the *principal directions*. The average of the two principal curvatures $H = \frac{1}{2}(\kappa_1 + \kappa_2)$ is called the *mean curvature*, and their product $K = \kappa_1\kappa_2$ is called the *Gaussian curvature*.

Surprisingly enough, though the principal curvatures are *extrinsic* quantities (i.e. quantities depending on the way the surface is embedded into the Euclidean space), the Gaussian curvature is an *intrinsic quantity*, that is, it can be fully expressed in terms of the metric of the surface. One of such definitions considers the perimeter $P(r)$ of a geodesic circle of radius r centered at a point $\mathbf{x}$ on the surface. On a Euclidean surface, $P(r) = 2\pi r$, while on curved surfaces a different quantity is measured. The defect of the perimeter is governed by the Gaussian curvature according to

$$K = \lim_{r \to 0} \frac{3(2\pi r - P(r))}{\pi r^3}. \tag{5.6}$$

5.2.3 Discrete Differential Geometry

The discretization of differential geometric quantities such as tangent and normal vectors, principal directions and curvatures, gradients, and the Laplace-Beltrami operator requires some attention, as straightforward differentiation with respect to some parametrization coordinates usually amplifies noise to unreasonable levels. In what follows, we briefly overview naïve and more robust methods for estimation of such quantities. The simplest discrete representation of a two-dimensional surface is a *point cloud* consisting of a set $\mathbb{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ of discrete points in $\mathbb{R}^3$ taken from the underlying continuous surface. Local connectivity information can be introduced by defining

an *edge set* $\mathbb{E} \subset \mathbb{X} \times \mathbb{X}$ indicating for each pair of samples $(\mathbf{x}, \mathbf{x}')$ in the cloud whether they are adjacent (i.e., $(\mathbf{x}, \mathbf{x}') \in E$) or not. This leads to an undirected graph representation. It is frequently convenient to approximate a continuous surface by a piecewise-planar one, consisting of a collection of polygonal patches glued together along their edges. A particular case of such polyhedral approximations are *triangular meshes* in which all faces are triangles built upon triples of points in the point cloud. Each triangle $\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k$ can be associated with the normal vector

$$\mathbf{N} = \frac{(\mathbf{x}_j - \mathbf{x}_i) \times (\mathbf{x}_k - \mathbf{x}_i)}{\|(\mathbf{x}_j - \mathbf{x}_i) \times (\mathbf{x}_k - \mathbf{x}_i)\|}. \tag{5.7}$$

The normal at a vertex of the mesh can be computed by averaging the normals to the triangles sharing that vertex, possibly weighted by the triangle areas. Such a neighborhood of a vertex is usually referred to as the 1-*ring*.

In the presence of noisy samples, the support of the 1-rings can be insufficient to reliably approximate the normal vectors. As an alternative, given a vertex $\mathbf{x}$, we can consider the r-neighborhood $\mathbb{N}_r(\mathbf{x}) = \mathbb{B}_r(\mathbf{x}) \cap \mathbb{X}$ where the ball $\mathbb{B}_r(\mathbf{x})$ is with respect to the Euclidean metric. The samples in $\mathbb{N}_r(\mathbf{x})$ can be further weighted inversely proportionally to their Euclidean distances from $\mathbf{x}$. The weighted second-order moment matrix

$$\mathsf{M} = \sum_{\mathbf{y} \in \mathbb{N}_r(\mathbf{x})} \alpha(\mathbf{y})(\mathbf{y} - \mathbf{x})(\mathbf{y} - \mathbf{x})^{\mathrm{T}} \tag{5.8}$$

represents the orientation of the surface in the vicinity of $\mathbf{x}$. Here, $\alpha(\mathbf{y})$ are assumed to be non-negative weights summing to one. The two largest eigenvectors $\mathbf{T}_1$ and $\mathbf{T}_2$ of M span the tangent plane $T_x\mathbb{X}$, while the third, smallest, eigenvector $\mathbf{N}$ corresponds to the normal. The tradeoff between sensitivity to small geometric features and robustness to noise is controlled by the local neighborhood radius r.

In the same way local that plane fitting constitutes a robust tool for the approximation of normals and tangents, fitting of a quadratic surface allows the estimation of second-order quantities related to curvature. For that purpose, the points in $\mathbb{N}_r(\mathbf{x})$ are first transformed so that $\mathbf{x}$ coincides with the origin, the z axis coincides with the normal, and the x and y axes coincide with the tangent vectors (i.e., a point y is represented as $\mathbf{y} = \mathbf{x} + u\mathbf{T}_1 + v\mathbf{T}_2 + w\mathbf{N}$). A paraboloid

$$w(u, v) = au^2 + buv + cv^2 \tag{5.9}$$

is then fit using weighted least squares. The Gaussian and mean curvatures can now be estimated using the closed-form expressions $K = 4ac - b^2$ and $H = a + c$; the principal curvatures and directions are obtained in a similar

way [9]. For a comprehensive overview of curvature discretization methods, the reader is referred to [52].

5.2.4 Diffusion Geometry

The positive semi-definite self-adjoint *Laplace-Beltrami operator* Δ_X associated with the metric tensor g is defined by the identity

$$\int f \Delta_X h \, d\text{vol} = -\int g_x(\nabla_X f, \nabla_X h) d\text{vol}, \tag{5.10}$$

which holds for any pair of smooth scalar fields $f, h : \mathbb{X} \to \mathbb{R}$;. Here, $d$vol denotes the differential area or volume element of the manifold, depending, respectively, whether the latter is 2D or 3D.

The Laplace-Beltrami operator can be expressed in the parametrization coordinates as

$$\Delta_X = -\frac{1}{\sqrt{\det g}} \sum_{ij} \partial_i \sqrt{\det g} g_{ij}^{-1} \partial_j, \tag{5.11}$$

When the metric is Euclidean ($g_{ij} = \mathbf{I}$), the operator reduces to the familiar

$$\Delta f = -\left(\frac{\partial^2 f}{\partial u_1^2} + \frac{\partial^2 f}{\partial u_2^2} \right) \tag{5.12}$$

(note that, in this chapter, we define the Laplacian with the minus sign in order to ensure its positive semi-definiteness).

The Laplace-Beltrami operator gives rise to the partial differential equation

$$\left(\frac{\partial}{\partial t} + \Delta_X \right) f(t, \mathbf{x}) = 0, \tag{5.13}$$

called the *heat equation*. The heat equation describes the propagation of heat on the surface and its solution $f(t, \mathbf{x})$ is the heat distribution at a point $\mathbf{x}$ in time t. The initial condition of the equation is some initial heat distribution $f(0, \mathbf{x})$; if $\mathbb{X}$ has a boundary, appropriate boundary conditions must be added. The solution of (5.13) corresponding to a point initial condition $f(0, \mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}')$, is called the *heat kernel* and represents the amount of heat transferred from $\mathbf{x}$ to $\mathbf{x}'$ in time t due to the diffusion process. Using spectral decomposition, the heat kernel can be represented as

$$k_t(\mathbf{x}, \mathbf{x}') = \sum_{i \geq 0} e^{-\lambda_i t} \phi_i(\mathbf{x}) \phi_i(\mathbf{x}'), \tag{5.14}$$

where ϕ_i and λ_i are, respectively, the eigenfunctions and eigenvalues of the Laplace-Beltrami operator satisfying $\Delta\phi_i = \lambda_i\phi_i$ (without loss of generality, we assume λ_i to be sorted in increasing order starting with $\lambda_0 = 0$). Since the Laplace-Beltrami operator is an *intrinsic* geometric quantity (i.e. it can be expressed solely in terms of the metric of $\mathbb{X}$), its eigenfunctions and eigenvalues as well as the heat kernel are invariant under isometric transformations of the manifold.

The value of the heat kernel $k_t(\mathbf{x}, \mathbf{x}')$ can be interpreted as the transition probability density of a random walk of length t from the point $\mathbf{x}$ to the point $\mathbf{x}'$. This allows the construction of a family of intrinsic metrics known as *diffusion metrics*,

$$\begin{aligned} d_t^2(\mathbf{x}, \mathbf{x}') &= \int \left(k_t(\mathbf{x}, \cdot) - k_t(\mathbf{x}', \cdot)\right)^2 d\mathrm{vol} \\ &= \sum_{i>0} e^{-\lambda_i t}(\phi_i(\mathbf{x}) - \phi_i(\mathbf{x}'))^2, \end{aligned} \tag{5.15}$$

which measure the "connectivity rate" of the two points by paths of length t.

The parameter t can be given the meaning of *scale*, and the family $\{d_t\}$ can be thought of as a scale-space of metrics. By integrating over all scales, a *scale-invariant* version of (5.15) is obtained,

$$\begin{aligned} d_{\mathrm{CT}}^2(\mathbf{x}, \mathbf{x}') &= 2\int_0^\infty d_t^2(\mathbf{x}, \mathbf{x}')dt \\ &= \sum_{i>0} \frac{1}{\lambda_i}(\phi_i(\mathbf{x}) - \phi_i(\mathbf{x}'))^2. \end{aligned} \tag{5.16}$$

This metric is referred to as the *commute-time distance* and can be interpreted as the connectivity rate by paths of any length. We will broadly call constructions related to the heat kernel, diffusion and commute time metrics as *diffusion geometry*.

5.2.5 Discrete Diffusion Geometry

The discretization of the heat kernel k_t and the associated diffusion geometry constructs is performed using formula (5.14), in which a finite number of eigenvalues and eigenfunctions of the discrete Laplace-Beltrami operator are taken. The latter can be computed directly using the finite elements method [70], of by discretization of the Laplace operator on the mesh followed by its eigendecomposition. For a manifold discretized as a vertex set $\mathbb{V}$, a discrete Laplace-Beltrami operator is expressed in the following generic form,

$$(\Delta_X f)_i = \frac{1}{a_i} \sum_j w_{ij}(f_i - f_j), \tag{5.17}$$

where w_{ij} are weights, a_i are normalization coefficients and the the elements $f_i = f(v_i)$ of the vector $\mathbf{f}$ represent a scalar function on the manifold (e.g. the heat distribution) sampled at the vertex set.

In matrix notation, (5.17) can be written as $\Delta_X \mathbf{f} = \mathsf{A}^{-1}\mathsf{W}\mathbf{f}$, where $\mathbf{f}$ is an $N \times 1$ vector, $\mathsf{A} = \mathrm{diag}(a_i)$ and $\mathsf{W} = \mathrm{diag}\left(\sum_{l \neq i} w_{il}\right) - (w_{ij})$. The discrete eigenfunctions and eigenvalues are found by solving the *generalized eigendecomposition* [48] $\mathsf{W}\Phi = \mathsf{A}\Phi\Lambda$, where $\Lambda = \mathrm{diag}(\lambda_l)$ is a diagonal matrix of eigenvalues and $\Phi = (\phi_l(v_i))$ is the matrix of the corresponding eigenvectors.

Different choices of A and W have been studied, depending on which continuous properties of the Laplace-Beltrami operator one wishes to preserve [32, 86]. For triangular meshes, a popular choice adopted in this chapter is the *cotangent weight* scheme [65, 57], in which

$$w_{ij} = \begin{cases} (\cot \alpha_{ij} + \cot \beta_{ij})/2 & (\mathbf{v}_i, \mathbf{v}_j) \in \mathbb{E}; \\ 0 & else, \end{cases} \tag{5.18}$$

where α_{ij} and β_{ij} are the two angles opposite to the edge between vertices $\mathbf{v}_i$ and $\mathbf{v}_j$ in the two triangles sharing the edge, and a_i are the discrete area elements (see Fig. 5.2).

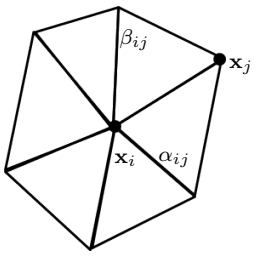


Fig. 5.2 One ring of a point $\mathbf{x}_i$, adjacent point $\mathbf{x}_j$ and the angles α_{ij} and β_{ij} used in the cotangent weight scheme to discretize the Laplace-Beltrami operator.

In many cases, the discretized Laplacian operator is only a means for the computation of its eigenfunctions and eigenvalues used to approximate the heat kernels. The family of finite elements methods (FEM) constitutes an alternative approach, allowing direct discretization of the eigendecomposition of the Laplace-Beltrami operator. Firstly, a set of M locally-supported basis functions $\{f_i(x)\}$ spanning a sufficiently smooth subspace of L^2 functions on the manifold is selected. Linear or cubic functions are the typical choice. An eigenfunction ϕ_i of the Laplacian admits the following weak form identity

$$\langle \nabla_X \phi_i, f_j \rangle = \lambda_i \langle \phi_i, f_j \rangle \tag{5.19}$$

for every basis function f_j. Expressing ϕ_i as a linear combination $\phi_i(x) \approx u_{i1} f_1(x) + \cdots + u_{iM} f_M(x)$, we arrive at the following system of equations:

$$\sum_{j=1}^{M} u_{ji} \langle \nabla_X f_i, f_j \rangle = \sum_{j=1}^{M} u_{ji} \lambda_i \langle f_i, f_j \rangle. \tag{5.20}$$

As the function α_i are fixed, the terms $a_{ij} = \langle \nabla_X f_i, f_j \rangle$ and $b_{ij} = \langle f_i, f_j \rangle$ are pre-computed forming the generalized eigendecomposition problem $\mathsf{AU} = \Lambda \mathsf{BU}$, where A and B are the $N \times M$ matrices with the elements a_{ij} and b_{ij}, respectively, and U is the $M \times M$ matrix with the elements u_{ij}. Once U is computed, the corresponding eigenfunctions can be found using the columns of U as the coefficients of linear combinations of the basis functions.

For volumetric data, the Laplace-Beltrami operator is usually discretized on a regular Cartesian grid using finite differences.

5.3 Feature Detectors

The goal of a feature detector is to find stable points or regions on a shape. The main requirements of a feature detector are that the points that it selects are (i) *repeatable*, that is, in two instances of a shape, ideally the same set of corresponding points is detected, and (ii) *informative*, that is, descriptors built upon these points contain sufficient information to distinguish the shape from others.

Since there is no single way to define a feature, the construction of the detector depends very much on the shape representation and the application at hand, or more specifically, the desired invariance properties.

5.3.1 A Taxonomy

In the most trivial case, no feature detection is performed and the feature descriptor is computed at all the points of the shape [17] or at some densely sampled subset thereof. The descriptor in this case is usually termed *dense*. Dense descriptors bypass the problem of repeatability, at the price of increased computational cost and potentially introducing many unimportant points that clutter the shape representation.

Many detectors assume to be given some scalar-valued or vector-valued function defined on the surface. The function can be either *photometric* information (texture) or a *geometric* quantity such as curvature. With this concept in mind, feature detection on shapes resembles very much that in images and many attempts to import methods from image processing and computer vision have been described in the literature. Several methods for

feature detection have been inspired by the *difference of Gaussians* (DOG), a classical feature detection approach used in computer vision. Zaharescu *et al.* [87] introduce the *mesh DOG* approach by first applying Gaussian filtering to scalar functions (e.g. mean or Gauss curvature) defined on the shape. This allows representation of the function in scale space and feature points are prominent maxima of the scale space across scales. Castellani *et al.* [21] apply Gaussian filtering directly on the mesh geometry and create a scale space describing the displacement of the mesh points in the normal direction.

A different class of feature detection methods tries to find stable components or regions in the analyzed shape. In the computer vision and image analysis community, stable component detection is used in the maximally stable extremal regions (MSER) algorithm [55]. MSER represents intensity level sets as a component tree and attempts to find level sets with the smallest area variation across intensity; the use of area ratio as the stability criterion makes this approach affine-invariant, which is an important property in image analysis, as it approximates viewpoint transformations. Methods similar to MSER have been explored in the works on topological persistence [30]. Persistence-based clustering [22] was used by Skraba *et al.* [76] to perform shape segmentation. In [27, 49] MSER-like features for meshes and volumes have been studied.

Because many feature detectors operate locally on a function defined on the shape, they are usually not very sensitive to non-rigid deformations. Nevertheless, there exist several geometric quantities based on the intrinsic properties of the manifold and thus theoretically invariant to isometric deformations by construction. Feature detectors based on such quantities are called *intrinsic* and also *isometry-invariant* or *bending-invariant*. Examples of intrinsic geometric quantities are the Gaussian curvature (which has been used in several settings of [87]), and heat kernels [81, 33]. Feature detection methods based on the heat kernel define a function on the shape, measuring the amount of heat remaining at a point $\mathbf{x}$ after large time t, given a point source at $\mathbf{x}$ at time 0, and detect features as local maxima of this function.

Another type of transformation, of interest in practical applications, is changes in topology, manifested as the presence of holes, missing parts, or changes in connectivity. Feature detectors insensitive to such changes (typically, a simpler case of point-wise connectivity change) are referred to as *topology-invariant*.

Table 5.3.1 summarizes the properties of known feature detectors, some of which are detailed in what follows.

[1] Unless truly intrinsic quantities are used.

Descriptor	Representation	Invariance			
		Scale	Rigid	Bending	Topology
Dense	Any	Yes	Yes	Yes	Yes
Harris 3D [74]	Any	No	Yes	Approx	Approx
Mesh DOG [87]	Mesh	No	Yes	Approx[1]	Approx
Salient features [21]	Mesh	No	Yes	Approx	Approx
Heat kernel [81]	Any	No	Yes	Yes	Approx
MSER [27, 49]	Mesh, volume	Yes	Yes	Yes	Approx

Table 5.1 Comparison of 3D feature detectors.

5.3.2 Harris 3D

An efficient feature detection method, called Harris operator, first proposed for the use in images [36] was extended to 3D shapes by Glomb [35] and Sipiran and Bustos [74]. This method is based on measuring variability of the shape in a local neighborhood of the point, by fitting a function to the neighborhood, and identifying feature points as points where the derivatives of this function are high [10]. Unlike images, 3D data might have arbitrary topology and sampling, which complicates the computation of derivatives.

For each point $\mathbf{x}$ on the shape, a neighborhood of radius ρ (typically, a k-ring in mesh representation) is selected. The neighborhood points are brought into a canonical system of coordinates by first subtracting the centroid. Next, a plane is fitted to the translated points by applying PCA and choosing the direction corresponding to the smallest eigenvalues as the direction of the normal. The points are rotated so that the normal is aligned with the z-axis. A quadratic function of the form $f(u,v) = \mathbf{a}^{\mathrm{T}}(u^2, uv, v^2, u, v, 1)$ is then fitted to the set of transformed points, yielding a parametric representation of the local extrinsic surface properties. Here, u and v denote the tangent plane coordinates, and $\mathbf{a}$ stands for the quadratic patch parameters.

A 2×2 symmetric matrix

$$\mathsf{E} = \frac{1}{\sqrt{2\pi\sigma}} \int_{\mathbb{R}^2} e^{-\frac{u^2+v^2}{2\sigma^2}} \begin{pmatrix} f_u^2(u,v) & f_u(u,v)f_v(u,v) \\ f_u(u,v)f_v(u,v) & f_v^2(u,v) \end{pmatrix} dudv \quad (5.21)$$

is computed. The *3D Harris operator* is defined as the map assigning $\mathsf{H}(\mathbf{x}) = \det(\mathsf{E}) - 0.04\mathrm{tr}^2(\mathsf{E})$ to each point $\mathbf{x}$ on the shape. A fixed percentage of points with the highest values of $H(\mathbf{x})$ are selected as the feature points.

In [74], the neighborhood radius ρ (alternatively, the k-ring width k) and the Gaussian variance σ are performed adaptively for each point in order to make the method independent on sampling and triangulation.

5.3.3 Mesh DOG

The Mesh DOG descriptor introduced in Zaharescu *et al.* [87] assumes the shape in mesh representation and in addition to be given some function f defined on the mesh vertices. The function can be either photometric information (texture) or a geometric quantity such as curvature.

Given a scalar function f on the shape, its *convolution* with a radially-symmetric kernel $k(r)$ is defined as

$$(f * k)(\mathbf{x}) = \int k(d(\mathbf{x}, \mathbf{y}))f(\mathbf{y})d\mathbf{y}, \tag{5.22}$$

where $d(\mathbf{x}, \mathbf{y})$ is the geodesic distance between points $\mathbf{x}$ and $\mathbf{y}$. Zaharescu *et al.* [87] propose the following r-ring approximation:

$$(f * k)(\mathbf{x}) = \frac{\sum_{\mathbf{y}\in\mathbb{N}_r(\mathbf{x})} k(\|\mathbf{x} - \mathbf{y}\|)f(\mathbf{y})}{\sum_{y\in\mathbb{N}_r(\mathbf{x})} k(\|\mathbf{x} - \mathbf{y}\|)}, \tag{5.23}$$

which assumes a uniformly sampled mesh. Here, $\mathbb{N}_r(\mathbf{x})$ denotes the r-ring neighborhood of the point $\mathbf{x}$.

By subsequently convolving a function f with a Gaussian kernel g_σ of width σ, a scale space $f_0 = f$, $f_k = f_{k-1} * g_\sigma$ is constructed. The *difference of Gaussians* (DOG) operator at scale k is defined as $DOG_k = f_k - f_{k-1}$.

Feature points are selected as the maxima of the DOG scale space across scales, followed by non-maximum suppression, using the one ring neighborhood in the current and the adjacent scales. A fixed percentage of points with the highest values of DOG are selected. To further eliminate unstable responses, only features exhibiting *corner-like* characteristics are retained. For this purpose, the *Hessian* operator at every point $\mathbf{x}$ is computed as

$$\mathsf{H} = \begin{pmatrix} f_{uu}(\mathbf{x}) & f_{uv}(\mathbf{x}) \\ f_{uv}(\mathbf{x}) & f_{vv}(\mathbf{x}), \end{pmatrix} \tag{5.24}$$

where f_{uu}, f_{uv} and f_{vv} are the second-order partial derivatives of f at $\mathbf{x}$. Second order derivatives are estimated w.r.t. some local system of coordinates u, v (e.g., by fixing $\mathbf{u}$ to be the direction of the gradient, $\mathbf{u} = \nabla_X f(\mathbf{x})$, and $\mathbf{v}$ perpendicular to it) by applying the directional derivative twice,

$$f_{uv}(\mathbf{x}) = \langle \nabla_X \langle \nabla_X f(\mathbf{x}), \mathbf{u} \rangle, \mathbf{v} \rangle. \tag{5.25}$$

The condition number $\lambda_{\max}/\lambda_{\min}$ of H (typically, around 10) is independent of the selection of the local system of coordinates and is used to threshold the features.

5.3.4 Salient Features

In Mesh DOG, the scale space is built by filtering a scalar function on the mesh while keeping the mesh geometry intact. Castellani *et al.* [21] proposed to create a scale space by filtering the shape itself.

Let $\mathbb{X}^0 = \{\mathbf{x}_i\} \subset \mathbb{R}^3$ denote the extrinsic coordinates of points on a surface. For example, if the shape is represented as a mesh, $\mathbb{X}^0$ is the point cloud comprising the mesh vertices. One can apply a Gaussian kernel g_σ to this point cloud, obtaining a new set of points $\mathbb{X}^1 = \mathbb{X}^0 * g_\sigma$, where

$$\mathbf{x}_i^1 = \sum_j g_\sigma(\|\mathbf{x}_i - \mathbf{x}_j\|)\mathbf{x}_j. \tag{5.26}$$

Applying the kernel several times creates a scale space of "blurred" shapes $\mathbb{X}^k$. One can naturally define a vector-valued DOG scale space on the original surface by assigning a vertex $\mathbf{x}_i$ the difference of the corresponding blurred coordinates, $\mathbf{d}_i^k = \mathbf{x}_i^k - \mathbf{x}_i^{k-1}$. By projecting $\mathbf{d}_i^k$ onto the normal $\mathbf{N}(\mathbf{x}_i)$ at the point $\mathbf{x}_i$, a scalar-valued DOG scale space, referred to as the *scale map* by the authors, is created. From this stage on, an approach essentially identical to Mesh DOG is undertaken. The authors do not use filtering by Hessian operator response and propose to use a robust method inspired by [37] to detect the feature points.

5.3.5 Heat Kernel Features

Recently, there has been increased interest in the use of *diffusion geometry* for shape recognition [71, 62, 56, 53, 15, 67]. In particular, the spectral decomposition (5.14) of the heat kernel is especially attractive as there exists efficient and stable methods to discretize the Laplace-Beltrami operator and its eigendecomposition.

The diagonal of the heat kernel at different scales, $k_t(\mathbf{x}, \mathbf{x})$, referred to as the *heat kernel signature* (HKS), can be interpreted as a multi-scale notion of the Gaussian curvature. Local maxima of the HKS for a large time parameter correspond to tips of protrusions that can be used as stable features as recently proposed by Sun *et al.* [81] and Gebal *et al.* [33].

In the simplest setting, feature points are found as two-ring local maxima of $k_t(\mathbf{x}, \mathbf{x})$ at a sufficiently large scale t [81]. In a more sophisticated setting, the persistence diagram of $k_t(\mathbf{x}, \mathbf{x})$ is computed, as described in the sequel, and features with insufficiently large distance between birth and death times are filtered out [76, 26, 10].

5.3.6 Topological Features

A different variety of feature-based techniques have been inspired by topological, rather than geometrical, shape analysis [61]. The most common tool used in applying topological methods to feature-based shape analysis is the notion of *topological persistence* introduced and formalized by Edelsbrunner et al. [30]. In its most basic form, topological persistence allows the definition of a pairing between critical values of a function defined on a topological domain (such as a simplicial complex) in a canonical way. This pairing defines a persistence value associated with each critical point, which provides a principled way of distinguishing prominent local maxima and minima from noise. Thus, these techniques fit naturally into the feature-based shape analysis framework, where both feature detection and description are often obtained via analysis of critical values of some function. Several techniques have been recently proposed for finding stable feature points by applying topological persistence to different functions defined on the shape, including the Heat Kernel Signature [76, 26] and the eigenfunctions of the Laplace-Beltrami operator [69].

Let w be some non-negative scalar function defined on the surface (e.g., the heat kernel $k_t(\mathbf{x}, \mathbf{x})$ for a moderate time scale t). A *super-level set* of w is defined as $\{\mathbf{x} : w(\mathbf{x}) \geq \tau\}$. By sweeping τ from zero to infinity, either new connected components of the level sets are born, or previously existing components merge. Each connected component can be associated with a local maximum of w, when the component is first born. Merging of two components corresponding to two local maxima $\mathbf{x}_1$ and $\mathbf{x}_2$ occurs at τ such that there exists a path connecting $\mathbf{x}_1$ and $\mathbf{x}_2$ along which $f(\mathbf{x}) \geq \tau$. We say that the component corresponding to a smaller local maximum $\mathbf{x}_1 < \mathbf{x}_2$ dies at time τ, that is, is merged into the component corresponding to the larger local maximum $\mathbf{x}_2$. The *persistence diagram* represents the births and deaths of all the connected components by assigning to each component a point in $[0, \infty]^2$, with the x and y coordinates representing the birth and the death time, respectively. The persistence of a local maximum is defined as the difference between its death and birth times. Maxima that never die have infinite persistence.

An excellent application of topological persistence to shape analysis and shape matching was demonstrated by Agarwal et al. [1], who used it to define a feature detector and descriptor, by defining a function on a surface, which approximately captures the concavity and convexity at each point in a parameter-free way. For every point $\mathbf{x}$ on the surface, the authors use topological persistence to find a canonical pair $\mathbf{y}$ which shares the normal direction with $\mathbf{x}$. Then the *elevation function* at $\mathbf{x}$ is simply the difference of the height values of $\mathbf{x}$ and $\mathbf{y}$ in this normal direction. Elevation function is invariant to rigid deformations and allows the analysis of both concavities and convexities in a unified fashion. Prominent minima and maxima of the elevation function can also be used as natural stable features of a shape. Persistent maxima of

the heat kernel $k_t(\mathbf{x},\mathbf{x})$ have also been shown to constitute robust and repeatable feature points [26]. Applying methods from computational topology to feature-based shape analysis is an active and potentially fruitful area of research and we refer the interested reader to a recent book [29].

5.3.7 Maximally Stable Components

Another class of methods, introduced in [27, 49], detects stable regions of the shape as an alternative to detecting stable points. For the discussion, we will assume that the shape is represented by a graph $G = (\mathbb{X}, \mathbb{E})$ weighted by a non-negative *vertex weight* $w : \mathbb{X} \to \mathbb{R}$ or *edge weight* $d : \mathbb{E} \to \mathbb{R}$. The graph is said to be *connected* if there exists a path between every pair of vertices in it. A graph $G' = (\mathbb{X}' \subseteq \mathbb{X}, \mathbb{E}' \subseteq \mathbb{E})$ is called a *subgraph* of G and denoted by $G' \subseteq G$. A maximal connected subgraph is called a *component* of $\mathbb{X}$. Given $\mathbb{E}' \subseteq \mathbb{E}$, the graph induced by $\mathbb{E}'$ is the graph $G' = (\mathbb{X}', \mathbb{E}')$ whose vertex set is made of all vertices belonging to an edge in $\mathbb{E}'$, that is, $\mathbb{X}' = \{\mathbf{x} \in \mathbb{X} : \exists \mathbf{x}' \in \mathbb{X}, (\mathbf{x}, \mathbf{x}') \in \mathbb{E}'\}$. Given a vertex-weighted graph (G, w) with a weighting function w, the *ℓ-cross-section* of G is defined as the graph induced by $E_\ell = \{(\mathbf{x}_1, \mathbf{x}_2) \in \mathbb{E} : w(\mathbf{x}_1), w(\mathbf{x}_2) \leq \ell\}$ for some $\ell \geq 0$. In the same way, a cross-section of an edge-weighted graph (G, d) is induced by the edge subset $E_\ell = \{e \in \mathbb{E} : d(e) \leq \ell\}$. A connected component of the cross-section is called an ℓ-level set of the weighted graph. The *altitude* of a component C is defined as the minimal ℓ for which C is a component of the ℓ-cross-section of G. Altitudes establish a partial order relation on the connected components of G as any component C is contained in a component with higher altitude. The set of all such pairs $(\ell(C), C)$ therefore forms a *component tree*.

Given a sequence $\{(\ell, C_\ell)\}$ of nested components forming a branch in the component tree, the *stability* of C_ℓ is defined as

$$s(\ell) = \frac{\mathrm{vol}(C_\ell)}{\frac{d}{d\ell}\mathrm{vol}(C_\ell)}, \tag{5.27}$$

where $\mathrm{vol}(C)$ denotes the area of the component C (or its volume in case of a three-dimensional manifold). In other words, the more the relative volume of a component changes with the change of ℓ, the less stable it is. A component C_{ℓ^*} is called *maximally stable* if the stability function has a local maximum at ℓ^*. Maximally stable components are widely known in the computer vision literature under the name of *maximally stable extremal regions* (*MSER*) [55], with $s(\ell^*)$ usually referred to as the region *score*. The construction of weighted component trees is based on the observation that the vertex set V can be partitioned into disjoint sets which are merged together going up in the tree. Maintaining and updating such a partition is done using the *union-*

find algorithm and related data structures with $\mathcal{O}(n \log n)$ complexity [60]. Such an approach is used to implement single-link agglomerative clustering which is adopted here for the construction of the component tree.

Fig. 5.3 Edge-weighted MSER. Figure reproduced from [49].

Using vertex-weighting, any scalar function that distinguishes between vertices and captures the local geometrical properties such as mean curvature [27] can be used. For non-rigid shape analysis, diffusion-geometric weights have a clear advantage being deformation-invariant [76] and easily computed through the heat kernel. The simplest vertex weight is obtained as the diagonal of the heat kernel $w(\mathbf{x}) = k_t(\mathbf{x}, \mathbf{x})$, which, up to a monotonic transformation, can be thought of as an approximation of the Gaussian curvature at t. The choice of the parameter t defines the scale of such an approximation [81]. A scale-invariant version of this weight (the *commute-time kernel*) is obtained by integrating k_t over all time scales in the range $[0, \infty]$,

$$w(\mathbf{x}) = \sum_{i=0}^{\infty} \lambda_i^{-1} \phi_i^2(\mathbf{x}). \tag{5.28}$$

Edge weights offer more flexibility allowing the expression of dissimilarity relations between adjacent vertices. Since the heat kernel $k_t(\mathbf{x}_1, \mathbf{x}_2)$ represents the proximity or "connectivity" of two vertices $\mathbf{x}_1, \mathbf{x}_2$, any function of the form $d(\mathbf{x}_1, \mathbf{x}_2) = \eta(k^{-1}(\mathbf{x}_1, \mathbf{x}_2))$ can define an edge weight inversely proportional to the heat kernel value (here η denotes a non-negative monotonic function). Another natural way of defining edge weights is the *diffusion distance* (5.15) or its scale-invariant version, the *commute time distance* (5.16).

While the original formulation of shape MSER detectors focused on meshes, in principle there exists no limitation to extend it to other representations as well. Recently, an extension of the method to volumetric data has been proposed, with the volumetric heat kernels and diffusion distances used as the weighting functions.

5.3.8 Benchmarks

An ideal feature detector should be repeatable under the desired class of shape transformations and also detect "rich" feature points from which informative descriptors can be drawn. While the latter is largely application and data-dependent, the repeatability of the detector can be evaluated quantitatively on a set of representative shape transformations. SHREC'10 robust feature detection and description benchmark [10] evaluates the detector repeatability by running the detector on a set of reference shapes. The detected features are used as reference locations. Then, detection is performed on the same shapes undergoing simulated transformations of different types (non-rigid bending, different types of noise, holes, etc.), for which groundtruth correspondence with the reference shapes is known. Repeatability is evaluated by counting the fraction of features that are consistently detected in the proximity of the reference locations. Different varieties of the heat kernel methods achieved the best results on this benchmark.

5.4 Feature Descriptors

Given a set of feature points (or, in the case of a dense descriptor, all the points on the shape), a local descriptor is then computed. There exists a plethora of different shape descriptor algorithms, which depend very much on the representation in which the shape is given (e.g. mesh or point cloud), the kind of information available and its quality (e.g. sampling resolution), and the application in mind. A descriptor can be characterized by its (i) "informativity", i.e., the information content that can be used for shape discrimination; (ii) invariance to some class of shape transformations, such as deformations or noise, (iii) computational complexity, (iv) compactness of descriptor and complexity of comparison of two descriptors (e.g. in shape matching applications). In addition, if the descriptor is used in combination with a feature detector, its sensitivity to feature location (detector repeatability) might be important. There are many tradeoffs between these properties that can be made in feature-based shape analysis applications.

5.4.1 A Taxonomy

Descriptors can be categorized as *geometric* or *photometric* (or both, referred to as *hybrid*), depending whether they rely only on the 3D geometry of the shape, or also make use of the texture. Some photometric descriptors can be adapted to work with geometric information, where some geometric property (e.g. curvature) is used in place of the texture [87]. A wide variety of geometric

quantities such as local patches [58], local moments [24] and volume [34], spherical harmonics [73], and contour and edge structures [63, 45] trying to emulate comparable features in images, can be used for geometric descriptors.

Multiscale descriptors (e.g. [81, 20, 68]) look at the shape at multiple levels of resolution, thus capturing different properties manifested at these scales. This is opposed to *single scale* or *scalar* descriptors, such as conformal factor [6].

Descriptors which are not altered by global scaling of the shape are called *scale-invariant*. Such an invariance can in some cases be achieved by shape normalization; a better approach is to build scale-invariance into the descriptor construction.

Because typically a descriptor operates locally around the feature point, feature descriptors are usually not very sensitive to non-rigid deformations of the shape. Nevertheless, there exist several geometric descriptors which are based on intrinsic properties of the manifold and thus theoretically invariant to isometric deformations by construction. Examples of intrinsic descriptors include histograms of local geodesic distances [66, 18], conformal factors [6], some settings of [87], and heat kernels [81, 20]. Such descriptors are called *intrinsic* and also *isometry-invariant* or *bending-invariant*.

Another type of transformation, of interest in practical applications, is changes in topology, manifested as the presence of holes, missing parts, or changes in connectivity. Descriptors insensitive to such changes (typically, a simpler case of point-wise connectivity change) are referred to as *topology-invariant*.

Finally, some authors [34] make a distinction between *high-dimensional* (or *rich*) and *low-dimensional* descriptors. The former refers to descriptors providing a fairly detailed description of the shape properties around the point such as [5, 39], while the latter computes only a few values per point and typically are curvature-like quantities such as *shape index* [42] and *curvedness* [43]. We find this division somewhat misleading, as there is no direct relation between the descriptor "richness" and dimensionality (recent works in computer vision on descriptor hashing and dimensionality reduction [79] demonstrate that rich descriptors such as SIFT can be compactly represented in much lower dimensions without losing much information). The question whether the "richness" of a descriptor is sufficient depends in general on the application and the data.

Table 5.2 summarizes the theoretical properties of known descriptors, some of which are detailed in what follows. The invariance properties of many descriptors were evaluated quantitatively in the SHREC robust feature detection and description benchmark [10], testing the descriptor variability under simulated transformations of different types (non-rigid bending, different types of noise, holes, etc.).

We devote particular attention in this section to different varieties of the recently introduced heat kernel signatures, which we consider to be one of the

most versatile descriptors currently available, possessing provable invariance properties, as well as a promising and interesting field for future research.

Descriptor	**Representation**	**Invariance**			
		Scale	Rigid	Bending	Topology
Gaussian curvature	Any	No	Yes	Yes	Approx[c]
Shape index [43]	Any	Yes	Yes	No	Approx[c]
Integral volume [34]	Volume, Mesh[a]	No	Yes	No	Approx[c]
Local histograms [66]	Any	No[b]	Yes	Yes	No[b]
HKS [81]	Any	No	Yes	Yes	Approx[c]
SIHKS [20]	Any	Yes	Yes	Yes	Approx[c]
CHKS [47]	Any+Texture	Yes	Yes	Yes	Approx[c,h]
VHKS [68]	Volume, Mesh[a]	No	Yes	Yes	Approx[c]
Spin image [39]	Any	No[i]	Yes	No[g]	Yes
Shape context [5]	Any	No	Yes	No	Yes
MeshHOG [87]	Mesh (+Texture)	Yes[d]	Yes	Approx[e]	Approx[d]
Conformal factor [6]	Mesh	No	Yes	Yes	No[f]

Table 5.2 Comparison of 3D feature descriptors. [a]Involving mesh rasterization. [b]Assuming geodesic distances. Different invariance properties can be achieved using diffusion or commute-time distances. [c]Point-wise connectivity changes have only a local effect and do not propagate to distant descriptors. [d]If photometric texture is used; in general, depending on the texture choice. [e]Triangulation-dependent. [f]Defined for shapes with fixed topology (e.g. watertight). [g]Can be made approximately invariant using small support. [h]The use of photometric information can reduce the sensitivity to topological noise compared to HKS. [i]Can be made scale invariance as in [44].

5.4.2 Curvature-based Descriptors (HK and SC)

The simplest and perhaps earliest shape descriptors based on curvature (also referred to as HK descriptors) were introduced by Besl [9, 8]. The combination of the mean curvature $H = \frac{1}{2}(\kappa_1 + \kappa_2)$ and the Gaussian curvature $K = \kappa_1 \cdot \kappa_2$ allow the classification of the type of a local surface patch as saddle valley ($K, H < 0$), saddle ridge ($K < 0, H > 0$), concave or convex cylinder ($K = 0, H < 0$ and $K = 0, H > 0$, respectively), concave or convex ellipsoid ($K > 0, H < 0$ and $K > 0, H > 0$, respectively), or plane ($K = H = 0$). The values of H and K depend on the shape scale.

Koenderink and van Doorn [43] defined a different descriptor (referred to as SC) which decouples the type and strength of local shape curvature as follows: The *shape index* $S = \frac{2}{\pi}\mathrm{atan}\left(\frac{\kappa_1+\kappa_2}{\kappa_1-\kappa_2}\right)$ is a scale-invariant continuous gradation of concave ($-1 < S < -0.5$), hyperbolic ($-0.5 < S < 0.5$) and convex ($0.5 < S < 1$) shapes. The *curvedness* $C = \sqrt{(\kappa_1^2 + \kappa_2^2)/2}$ measures how strong the curvature of a particular local shape type is at a point. Planar shapes

have indeterminate shape index and can be determined from the curvedness $C = 0$.

Both the HK and SC descriptors make use of the mean curvature, which is not intrinsic and hence not deformation invariant.

5.4.3 Spin Images

The *spin image* descriptor [39, 2, 3] represents the neighborhood of a point on a shape by fitting an oriented coordinate system at the point. The local system of cylindrical coordinates at point $\mathbf{x}$ is defined using the normal and tangent plane: the radial coordinate α defined as the perpendicular distance to the line through the surface normal $\mathbf{n}(\mathbf{x})$, and the elevation coordinate β, defined as the signed perpendicular distance to the tangent plane. The cylindrical angular coordinate is omitted because it cannot be defined robustly and unambiguously on many surface patches, such as those where the curvature is the similar in all directions.

A spin image is a histogram of points in the *support region* represented in α, β coordinates. The bins can be in linear or logarithmic scale. The support region is defined by limiting the range of the values of α and β (thus looking at points $\mathbf{y}$ within some distance from $\mathbf{x}$) and requiring that $\cos^{-1}\langle \mathbf{n}(\mathbf{x}), \mathbf{n}(\mathbf{y}) \rangle < \epsilon$ (limiting self occlusion artifacts). The histogram can be represented as a 2D image, hence the name of the descriptor (Fig. 5.4).

The spin image is applicable to any shape representation in which the point coordinates are explicitly given and normals and tangent planes can be computed (e.g., meshes or point clouds). Because of dependence on the embedding coordinates, such a descriptor is not deformation-invariant.

5.4.4 Shape Context

The concept of the *shape context* descriptor was first introduced in [5] for image analysis, though it is directly applicable to 3D shapes [46]. The shape context describes the structure of the shape as relations between a point to the rest of the points. Given the coordinates of a point $\mathbf{x}$ on the shape, the shape context descriptor is constructed as a histogram of the direction vectors from $\mathbf{x}$ to the rest of the points, $\mathbf{y} - \mathbf{x}$. Typically, a log-polar histogram is used. The descriptor is applicable to any shape representation in which the point coordinates are explicitly given, such as mesh, point cloud, or volume. Because of dependence on the embedding coordinates, such a descriptor is not deformation-invariant.

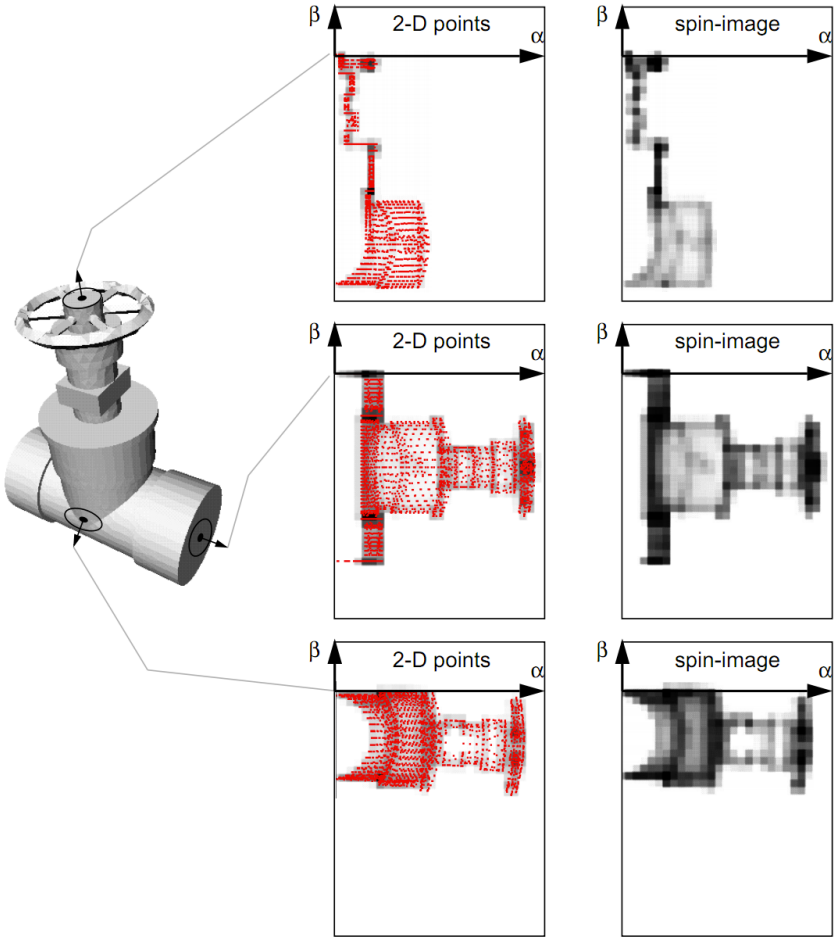


Fig. 5.4 Example of spin image descriptor computation (image taken from [38]).

5.4.5 Integral Volume Descriptor

The *integral volume* descriptor, used in [34], is an extension to 3D shapes of the concept of integral invariants introduced for image description in [54]. Given a solid object Ω with a boundary $X = \partial\Omega$, the descriptor measures volume contained in a ball of fixed radius r,

$$V_r(\mathbf{x}) = \int_{B_r(\mathbf{x})\cap\Omega} d\mathbf{x}. \tag{5.29}$$

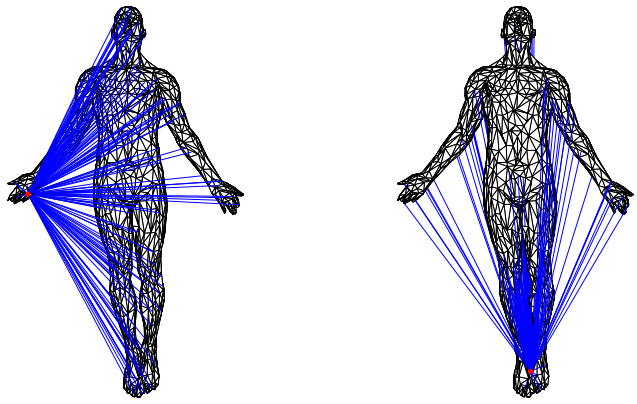


Fig. 5.5 Example of shape context computation. Shown in red is the reference point $\mathbf{x}$, and in blue the rays $\mathbf{y} - \mathbf{x}$.

If $B_r(\mathbf{x}) \cap \Omega$ is simply connected, the volume descriptor can be related to the mean curvature $H(\mathbf{x})$ as $V_r(\mathbf{x}) = \frac{2\pi}{3} r^3 - \frac{\pi}{4} H r^4 + \mathcal{O}(r^5)$ [34]. Since the mean curvature is not intrinsic, the descriptor is sensitive to deformations of the shape. Varying the value of r, a multi-scale descriptor can be computed. Numerically, the descriptor is efficiently computed in a voxel representation of the shape, by means of convolution with the ball mask.

5.4.6 Mesh Histogram of Gradients (HOG)

MeshHOG [87] is a shape descriptor emulating SIFT-like image descriptors [51], referred to as *histograms of gradients* or *HOG*. The descriptor assumes the shape in mesh representation and in addition to be given some function f defined on the mesh vertices. The function can be either photometric information (texture) or a geometric quantity such as curvature. The descriptor at point $\mathbf{x}$ is computed by creating a local histogram of gradients of f in an r-ring neighborhood of $\mathbf{x}$. The gradient ∇f is defined extrinsically as a vector in $\mathbb{R}^3$ but projected onto the tangent plane at $\mathbf{x}$ which makes it intrinsic. The descriptor support is divided into four polar slices (corresponding to 16 quadrants in SIFT). For each of the slices, a histogram of 8 gradient orientations is computed. The result is a 32-dimensional descriptor vector obtained by concatenating the histogram bins.

The MeshHOG descriptor works with mesh representations and can work with photometric or geometric data or both. It is intrinsic in theory, though the specific implementation in [87] depends on triangulation.

5.4.7 Heat Kernel Signature (HKS)

The *heat kernel signature* (HKS) was proposed in [81] as an intrinsic descriptor based on the properties of heat diffusion and defined as the diagonal of the heat kernel. Given some fixed time values $t_1, \ldots, t_n$, for each point x on the shape, the HKS is an n-dimensional descriptor vector

$$p(\mathbf{x}) = (k_{t_1}(\mathbf{x}, \mathbf{x}), \ldots, k_{t_n}(\mathbf{x}, \mathbf{x})). \tag{5.30}$$

Intuitively, the diagonal values of the heat kernel indicate how much heat remains at a point after certain time (or alternatively, the probability of a random walk to remain at a point if resorting to the probabilistic interpretation of diffusion processes) and is thus related to the "stability" of a point under diffusion process.

The HKS descriptor is intrinsic and thus isometry-invariant, captures local geometric information at multiple scales, is insensitive to topological noise, and is informative (if the Laplace-Beltrami operator of a shape is non-degenerate, then any continuous map that preserves the HKS at every point must be an isometry). Since the HKS can be expressed in the Laplace-Beltrami eigenbasis as

$$k_t(\mathbf{x}, \mathbf{x}) = \sum_{i \geq 0} e^{-t\lambda_i} \phi_i^2(\mathbf{x}), \tag{5.31}$$

it is easily computed across different shape representations for which there is a way to compute the Laplace-Beltrami eigenfunctions and eigenvalues.

5.4.8 Scale-invariant Heat Kernel Signature (SI-HKS)

A disadvantage of the HKS is its dependence on the global scale of the shape. If $\mathbb{X}$ is globally scaled by β, the corresponding HKS is $\beta^{-2} k_{\beta^{-2}t}(\mathbf{x}, \mathbf{x})$. In some cases, it is possible to remove this dependence by *global* normalization of the shape.

A *scale-invariant HKS* (SI-HKS) based on *local* normalization was proposed in [20]. Firstly, the heat kernel scale is sampled logarithmically with some basis α, denoted here as $k(\tau) = k_{\alpha^\tau}(x, x)$. In this scale-space, the heat kernel of the scaled shape becomes $k'(\tau) = a^{-2} k(\tau + 2 \log_\alpha a)$ (Fig. 5.6a). Secondly, in order to remove the dependence on the multiplicative constant a^{-2},

the logarithm of the signal followed by a derivative w.r.t. the scale variable is taken,

$$\begin{aligned}\frac{d}{d\tau}\log k'(\tau) &= \frac{d}{d\tau}\left(-2\log a + \log k(\tau + 2\log_\alpha a)\right)\\ &= \frac{d}{d\tau}\log k(\tau + 2\log_\alpha a)\\ &= \frac{\frac{d}{d\tau}k(\tau + 2\log_\alpha a)}{k(\tau + 2\log_\alpha a)}.\end{aligned} \tag{5.32}$$

Denoting

$$\tilde{k}(\tau) = \frac{\frac{d}{d\tau}k(\tau)}{h(\tau)} = \frac{-\sum_{i\geq 0}\lambda_i \alpha^\tau \log\alpha e^{-\lambda_i\alpha^\tau}\phi_i^2(\mathbf{x})}{\sum_{i\geq 0} e^{-\lambda_i\alpha^\tau}\phi_i^2(\mathbf{x})}, \tag{5.33}$$

one thus has a new function $\tilde{k}$ which transforms as $\tilde{k}'(\tau) = \tilde{k}(\tau + 2\log_\alpha a)$ as a result of scaling (Fig. 5.6b). Finally, by applying the Fourier transform to $\tilde{k}$, the shift becomes a complex phase,

$$\mathcal{F}[\tilde{k}'](\omega) = \tilde{\mathcal{K}}'(\omega) = \tilde{\mathcal{K}}(\omega)e^{-j\omega 2\log_\alpha a}, \tag{5.34}$$

and taking the absolute value in the Fourier domain (Fig. 5.6c),

$$|\tilde{\mathcal{K}}'(\omega)| = |\tilde{\mathcal{K}}(\omega)|, \tag{5.35}$$

produces a scale-invariant descriptor (Fig. 5.7).

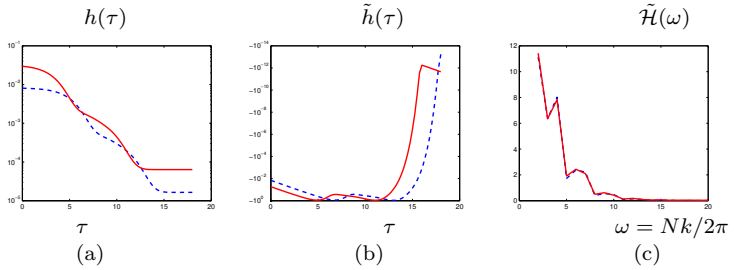


Fig. 5.6 Construction of the Scale-Invariant HKS: (a) we show the HKS computed at the same point, for a shape that is scaled by a factor of 11 (blue dashed plot); please notice the log-scale. (b) The signal $\tilde{h}(\tau)$, where the change in scale has been converted into a shifting in time. (c) The first 10 components of $|\tilde{\mathcal{H}}(\omega)|$ for the two signals; the descriptors computed at the two different scales are virtually identical.

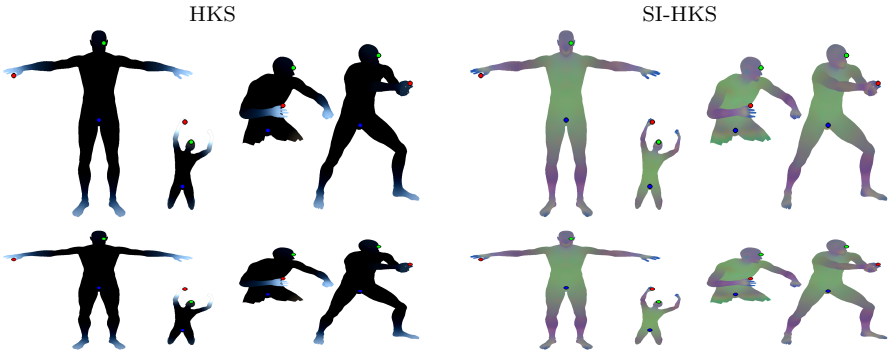


Fig. 5.7 Top: three components of the HKS (left) and the proposed SI-HKS (right), represented as RGB color and shown for different shape transformations (null, isometric deformation+scale, missing part, topological transformation). Bottom: HKS (left) and SI-HKS (right) descriptors at three points of the shape (marked with red, green, and blue). Dashed line shows the null shape descriptor.

5.4.9 Color Heat Kernel Signature (CHKS)

If, in addition, *photometric* information is available, given in the form of *texture* $\alpha : X \to \mathcal{C}$ in some m-dimensional colorspace $\mathcal{C}$ (e.g. $m = 1$ in case of grayscale texture and $m = 3$ in case of color texture), it is possible to design diffusion processes that take into consideration not only geometric but also photometric information [77, 47]. For this purpose, let us assume the shape X to be a submanifold of some $(m + 3)$-dimensional manifold $\mathcal{E} = \mathbb{R}^3 \times \mathcal{C}$ with the Riemannian metric tensor g, embedded by means of a diffeomorphism $\xi : X \to \xi(X) \subseteq \mathcal{E}$. A Riemannian metric on the manifold X induced by the embedding is the *pullback metric* $(\xi^* g)(r, s) = g(d\xi(r), d\xi(s))$ for $r, s \in T_x X$, where $d\xi : T_x X \to T_{\xi(x)}\mathcal{E}$ is the differential of ξ, and T denotes the tangent space. In coordinate notation, the pullback metric is expressed as $(\xi^* g)_{\mu\nu} = g_{ij}\partial_\mu \xi^i \partial_\nu \xi^j$, where the indices $i, j = 1, \ldots, m + 3$ denote the embedding coordinates.

The structure of $\mathcal{E}$ is to model joint geometric and photometric information. The geometric information is expressed by the embedding coordinates $\xi_g = (\xi^1, \ldots, \xi^3)$; the photometric information is expressed by the embedding coordinates $\xi_p = (\xi^4, \ldots, \xi^{3+m}) = (\alpha^1, \ldots, \alpha^m)$. In a simple case when $\mathcal{C}$ has a Euclidean structure (for example, the Lab colorspace has a natural Euclidean metric), the pullback metric boils down to $(\xi^* g)_{\mu\nu} = \langle \partial_\mu \xi_g, \partial_\nu \xi_g \rangle_{\mathbb{R}^3} + \langle \partial_\mu \xi_p, \partial_\nu \xi_p \rangle_{\mathbb{R}^3}$.

The Laplace-Beltrami operator associated with such a metric gives rise to a heat diffusion equation which takes into consideration both the geometry and the color of the object (simplistically put, the heat flows more slowly

across different colors). This, in turn, allows the definition of a color-sensitive HKS (CHKS) that merges geometric and photometric information [47].

5.4.10 Volumetric Heat Kernel Signature (VHKS)

So far, we have considered the shape as a 2D boundary surface of a 3D physical object and represented the deformations of the object as deformation of the 2D surface. While physically realistic transformations of reasonably inelastic objects can be modeled as isometries of the 2D boundary surface ("boundary isometries"), the converse is not true: one can easily think of a transformation that preserves the metric structure of the object boundary, but not the volume (Fig. 5.8). Such transformations are not physically realistic, as they change the object volume or mass. However, all intrinsic descriptors we have discussed (including the HKS) would be invariant to such boundary isometries and thus have "too much invariance".

A different approach is to consider shapes as volumes and require invariance to transformations that preserve the metric structure inside the volume ("volume isometries"). Such descriptors are called *volumetric*. The idea of the heat kernel descriptor can be applied to volumetric shape representations [68]. In this case, given a solid object Ω, the heat diffusion inside the volume is given by the heat equation with Neumann boundary conditions on the boundary $\partial\Omega$,

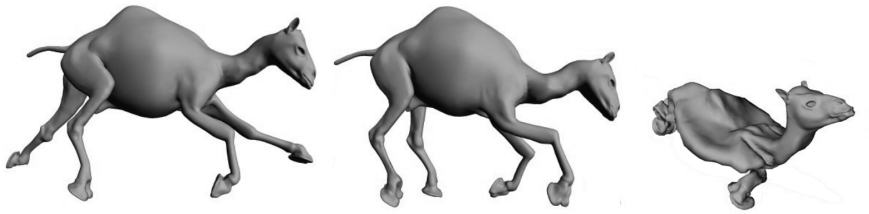
$$\begin{aligned} \left(\Delta + \frac{\partial}{\partial t}\right) U(\mathbf{x}, t) = 0 \quad & \mathbf{x} \in \mathrm{int}(\Omega), \\ \langle \nabla U(\mathbf{x}, t), \mathbf{n}(\mathbf{x}) \rangle = 0 \quad & \mathbf{x} \in \partial\Omega \end{aligned} \tag{5.36}$$

where $\mathbf{n}$ is the normal to the boundary surface $\partial\Omega$, Δ is the positive-semidefinite Laplacian operator in $\mathbb{R}^3$, and $U : \Omega \times [0, \infty) \to \mathbb{R}$ is the volumetric heat distribution in Ω. The *volumetric heat kernel signature* (VHKS) is defined as the diagonal of the heat kernel of (5.36) at a set of time values t, expressible in the eigenbasis of the Laplacian as

$$K_t(\mathbf{x}, \mathbf{x}) = \sum_{l=0}^{\infty} e^{-\Lambda_l t} \Phi_l(\mathbf{x})^2, \tag{5.37}$$

where Λ_l, Φ_l are the eigenvalues and eigenfunctions of the Laplacian operator with the above boundary conditions,

$$\begin{aligned} \Delta\Phi_l(\mathbf{x}) &= \Lambda_l \Phi_l(\mathbf{x}); \\ \langle \nabla\Phi_l(\mathbf{x}), \mathbf{n}(\mathbf{x}) \rangle = 0 \quad & \mathbf{x} \in \partial\Omega. \end{aligned} \tag{5.38}$$



(a) (b) (c)

Fig. 5.8 Volumetric (a,b) and boundary (c) isometric deformations of a camel shape. Figure from [80]. According to [68], volume isometries are a better model of physical objects deformation than boundary isometries.

The descriptor can be computed on any volumetric representation of the shape, allowing for efficient computation of the Laplacian eigenvalues and eigenfunctions. For meshes and other surface representations, it is necessary to perform rasterization to convert them into voxel representation [68].

The VHKS is invariant to volumetric isometries of the shape (i.e. deformations that do not change the metric structure inside the shape). Such transformations are necessarily isometries of the boundary $\partial\Omega$, but not vice versa. Thus, VHKS does not have the extra invariance that HKS has.

5.4.11 Case Study: Shape Retrieval using Two Heat Kernel Descriptors (HKS and SI-HKS)

We conclude this section with a case study comparing the performance of two descriptors on a shape retrieval application using a "bags of features". For additional details on this application, the reader is referred to Chap. 7. A bag of features is a histogram of vector-quantized descriptors, which allows an efficient computation of similarity between shapes, boiling down to the comparison of two histograms. The bag of features inherits the invariance of the underlying local feature descriptor used for its construction. Thus, the choice of the descriptor is crucial for obtaining desired invariance.

In this test case, we used the cotangent weight scheme to discretize the surface Laplace-Beltrami operator; the heat kernel h was approximated using $k = 100$ largest eigenvalues and eigenvectors. For HKS, we used six scales $1024, 1351, 1783, 2353, 3104$ and 4096; for the SI-HKS, we used an exponential scale-space with base $\alpha = 2$ and τ ranging from 1 to 25 with increments of $1/16$. After applying logarithm, derivative, and Fourier transform, the first 6 lowest frequencies were used as the local descriptor. SHREC 2010 dataset was used. The query set consisted of shapes from the dataset undergoing different transformations. Only a single correct match exists in the database, and ideally, it should be the first one.

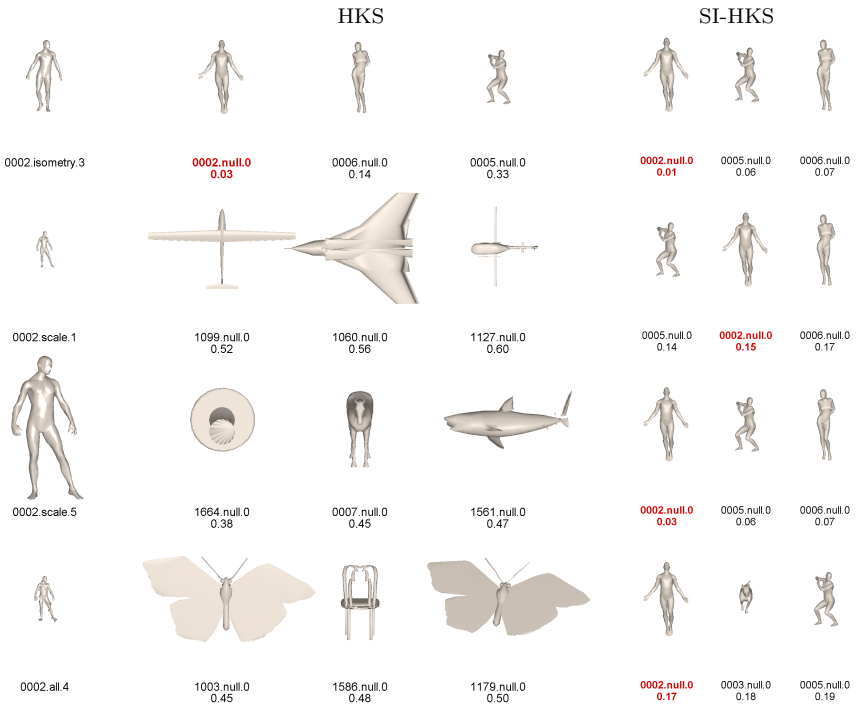


Fig. 5.9 Retrieval results using bags of features computed with HKS and SIHKS, tested on the SHREC'10 robust large-scale shape retrieval dataset. Left: query shapes, middle: first three matches obtained with HKS descriptors, right: first three matches obtained with SI-HKS descriptors. Only a single correct match exists in the database (marked in red), and ideally, it should be the first one.

An example of retrieval using bags of features built from HKS and SI-HKS descriptors is shown in Fig. 5.9. It clearly shows the failure of HKS-based bags of features to correctly find similarity between scaled shapes, which makes the use of SI-HKS preferable over HKS in problems involving arbitrary shape scaling.

5.5 Research Challenges

Current challenges in descriptor research include finding a good proportion between theoretical invariance, discriminativity, sensitivity to noise and computational complexity. A single ideal tradeoff is unlikely to be found, since these parameters heavily depend on the application.

Another important challenge of interest both in 2D image and 3D shape analysis is incorporating spatial relations between features. Most feature de-

scriptors capture only local information of the shape, while it is known that, in many cases, the spatial relations between different features are not less important. For example, on a human hand one would find five similar features corresponding to fingers, while their particular spatial configuration is what makes the object recognizable as a hand. Taking this example *ad absurdum*, one can think of a "soup of features" which have no clear spatial configuration. Recent works on symmetry [40, 59, 66, 62, 67], self-similarity [64, 16] and structure detection is a step in this direction.

5.6 Conclusions

This chapter described feature-based methods in 3D shape analysis that are commonly used in applications such as content-based shape retrieval and invariant shape correspondence. Though not completely new, this field is relatively unexplored in shape analysis and has lagged behind similar methods in 2D image analysis. The success of feature-based methods in computer vision has recently brought significant interest for developing similar approaches in shape analysis, either by finding analogies to 2D image features, or by exploiting 3D shape-specific construction.

This chapter tried to overview the basic ideas and principles in modern feature-based shape analysis models, placing an emphasis on feature detection and description, their invariance and quality. For different applications of these methods (e.g. in shape retrieval), the reader is referred to other chapters in this book.

It should be understood that feature-based shape analysis is a complicated pipeline with inter-dependent stages and, in particular, feature detection and description is a linked pair of operations. In some cases, the link is very natural and homogenous and falls under the same mathematical model. For example, heat kernels can be used for both detection (heat kernel features) and description (heat kernel signatures).

More generically, it is possible to use different methods for feature detection and description. However, it is important to understand the strengths and limitations of each approach, which are often dependent on the data in hand. For example, one might use Gaussian curvature extrema as a detector, but then using spin images as a descriptor could be a bad idea, since they depend on the normal and Gaussian curvature extrema are exactly where this normal is changing the most.

5.7 Further Reading

For a broad overview of geometric foundations and algorithms in shape analysis, we refer the reader to [14]. For an excellent comprehensive treatment of differential and Riemannian geometry the reader is referred to [28] and [7]. While focusing on *local* features in this chapter, we intentionally completely left aside an interesting and broad field on *global* or *holistic* shape descriptors based on global geometric quantities such as metric distributions. The reader is referred to the chapter on manifold intrinsic similarity in [72] for an overview of these methods. An introduction to diffusion geometry and diffusion maps can be found in [25]. Details of SHREC'10 benchmarks mentioned in this chapter appear in [10, 11, 12]. Algorithms for dimensionality reduction and hashing of descriptors in the context of deformable shape retrieval are discussed in [19, 79]. For additional details on shape retrieval, refer to Chap. 7 of this book.

5.8 Questions

1. Give three example applications where feature-based methods in 3D shape analysis are useful.
2. What qualities do we look for in a good feature detector?
3. What qualities do we look for in a good feature descriptor?
4. Explain why descriptor invariance and the information richness of a descriptor often need to be traded-off against each other. Use spin images as an example in your argument.
5. Explain why Gaussian curvature is an intrinsic shape property, whereas mean curvature is not.
6. Outline how heat kernel signatures can be used to determine a set of correspondences across two shapes.
7. Explain the effect of global uniform shape scaling of the shape on the eigenfunctions and eigenvalues of the Laplace-Beltrami operator.

5.9 Exercises

1. Compute the Gaussian curvature map on a 3D shape and display it as a colormap. Which points have positive Gaussian curvature? Negative curvature?
2. Repeat Exercise 1 using several local scales. How does it affect the result?
3. Determine the local extrema of the Gaussian curvature at the most suitable scale. Display the detected points by overlaying them on the 3D shape.

4. Compute a set of descriptors at the points detected in Exercise 3. As the descriptor at each point, use the coordinates x, y, z. Under what circumstances is x, y, z a good/poor choice of descriptor with this particular detector?
5. Match the descriptors across a pair of facial scans of the same person using simple Euclidean metric. Note the problems matching across the symmetrical face. Suggest solutions to this problem of point matching for symmetrical objects.
6. Given a shape represented as a triangular mesh, discretize the Laplace-Beltrami operator using the cotangent weights formula. Compute and show the eigenvalues and eigenfunctions of the Laplace-Beltrami operator.
7. Test the effect of shape transformations on the eigenvalues and eigenfunctions of the Laplace-Beltrami operator. Experimentally confirm their bending-invariance.
8. Using the eigenvalues and eigenfunctions of the Laplace-Beltrami operator, compute the heat kernel. Show the heat kernel diagonal values at points of different curvature.
9. Using the heat kernel diagonal, compute the HKS descriptor. Test its behavior for different values of time scale.
10. Compute dense HKS descriptor on multiple shapes (use for example the SHREC dataset). Create a geometric vocabulary by clustering in the HKS descriptor space. Compute a bag of features by means of vector quantization of the descriptors in the geometric vocabulary. Compare different settings of hard and soft quantization.
11. Using bags of features computed in Exercise 10, perform shape retrieval on a dataset of shapes undergoing deformations. Experimentally observe invariance and sensitivity of the descriptors.

References

1. P. K. Agarwal, H. Edelsbrunner, J. Harer, and Y. Wang, *Extreme elevation on a 2-manifold*, SCG '04: Proceedings of the twentieth annual symposium on Computational geometry, 2004, pp. 357–365.
2. M. Andreetto, N. Brusco, and G. M Cortelazzo, *Automatic 3D modeling of textured cultural heritage objects*, Trans. Image Processing **13** (2004), no. 3, 335–369.
3. J. Assfalg, M. Bertini, and P. Pala A. Del Bimbo, *Content-based retrieval of 3d objects using spin image signatures*, Trans. Multimedia **9** (2007), no. 3, 589–599.
4. H. Bay, T. Tuytelaars, and L. Van Gool, *SURF: Speeded up robust features*, Proc. ECCV, 2006, pp. 404–417.
5. S. Belongie, J. Malik, and J. Puzicha, *Shape matching and object recognition using shape contexts*, Trans. PAMI (2002), 509–522.
6. M. Ben-Chen, O. Weber, and C. Gotsman, *Characterizing shape using conformal factors*, Proc. 3DOR, 2008.
7. M. Berger, *A panoramic view of Riemannian geometry*, Springer Verlag, 2003.
8. P. J. Besl, *Surfaces in range image understanding*, Springer-Verlag, 1988.

9. P. J. Besl and R. C. Jain, *Invariant surface characteristics for 3d object recognition in range images*, Computer vision, graphics, and image processing **33** (1986), no. 1, 33–80.
10. A. M. Bronstein, M. M. Bronstein, B. Bustos, U. Castellani, M. Crisani, B. Falcidieno, L. J. Guibas, I. Isipiran, I. Kokkinos, V. Murino, M. Ovsjanikov, G. Patané, M. Spagnuolo, and J. Sun, *Shrec 2010: robust feature detection and description benchmark*, Proc. 3DOR, 2010.
11. A. M. Bronstein, M. M. Bronstein, U. Castellani, B. Falcidieno, A. Fusiello, A. Godil, L. J. Guibas, I. Kokkinos, Z. Lian, M. Ovsjanikov, G. Patané, M. Spagnuolo, and R. Toldo, *Shrec 2010: robust large-scale shape retrieval benchmark*, Proc. 3DOR, 2010.
12. A. M. Bronstein, M. M. Bronstein, U. Castellani, A. Dubrovina L. J. Guibas, R. P. Horaud, R. Kimmel, D. Knossow, E. von Lavante, D. Mateus, M. Ovsjanikov, and A. Sharma, *Shrec 2010: robust correspondence benchmark*, Proc. 3DOR, 2010.
13. A. M. Bronstein, M. M. Bronstein, and R. Kimmel, *Three-dimensional face recognition*, Int'l J. Computer Vision (IJCV) **64** (2005), no. 1, 5–30, 3D face recognition using metric model.
14. A. M. Bronstein, M. M. Bronstein, and R. Kimmel, *Numerical geometry of non-rigid shapes*, Springer-Verlag New York, 2008, First systematic treatment of non-rigid shapes.
15. A. M. Bronstein, M. M. Bronstein, R. Kimmel, M. Mahmoudi, and G. Sapiro, *A Gromov-Hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching*, IJCV (2010).
16. A. M. Bronstein, M. M. Bronstein, and N. J. Mitra, *Intrinsic regularity detection in 3D geometry*, Proc. ECCV, 2010.
17. A. M. Bronstein, M. M. Bronstein, M. Ovsjanikov, and L. J. Guibas, *Shape google: a computer vision approach to invariant shape retrieval*, Proc. NORDIA, 2009.
18. A.M. Bronstein, M.M. Bronstein, A.M. Bruckstein, and R. Kimmel, *Partial similarity of objects, or how to compare a centaur to a horse*, International Journal of Computer Vision **84** (2009), no. 2, 163–183.
19. A.M. Bronstein, M.M. Bronstein, L.J. Guibas, and M. Ovsjanikov, *Shape Google: Geometric words and expressions for invariant shape retrieval*, ACM Transactions on Graphics (TOG) **30** (2011), no. 1.
20. M. M. Bronstein and I. Kokkinos, *Scale-invariant heat kernel signatures for non-rigid shape recognition*, Proc. CVPR, 2010.
21. U. Castellani, M. Cristani, S. Fantoni, and V. Murino, *Sparse points matching by combining 3D mesh saliency with statistical descriptors*, Computer Graphics Forum **27** (2008), 643–652.
22. F. Chazal, L.J. Guibas, S. Oudot, and P. Skraba, *Persistence-based clustering in Riemannian manifolds*, (2009).
23. O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, *Total recall: Automatic query expansion with a generative feature model for object retrieval*, Proc. ICCV, 2007.
24. U. Clarenz, M. Rumpf, and A. Telea, *Robust feature detection and local classification for surfaces based on moment analysis*, Trans. Visualization and Computer Graphics **10** (2004), no. 5, 516–524.
25. R.R. Coifman and S. Lafon, *Diffusion maps*, Applied and Computational Harmonic Analysis **21** (2006), no. 1, 5–30, Definition of diffusion distance.
26. T. K. Dey, K. Li, C. Luo, P. Ranjan, I. Safa, and Y. Wang, *Persistent Heat Signature for Pose-oblivious Matching of Incomplete Models*, Proc. SGP, 2010, pp. 1545–1554.
27. J. Digne, J.M. Morel, N. Audfray, and C. Mehdi-Souzani, *The level set tree on meshes*, (2010).
28. M.P. Do Carmo, *Differential geometry of curves and surfaces*, vol. 2, Prentice-Hall Englewood Cliffs, NJ, 1976.
29. H. Edelsbrunner and J. Harer, *Computational Topology. An Introduction*, Amer. Math. Soc., Providence, Rhode Island, 2010.

30. H. Edelsbrunner, D. Letscher, and A. Zomorodian, *Topological persistence and simplification*, Discrete and Computational Geometry **28** (2002), no. 4, 511–533.
31. A. Elad and R. Kimmel, *On bending invariant signatures for surfaces*, IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI) **25** (2003), no. 10, 1285–1295, Introduction of canonical forms.
32. M. S. Floater and K. Hormann, *Surface parameterization: a tutorial and survey*, Advances in Multiresolution for Geometric Modelling **1** (2005).
33. K. Gebal, J. A. Bærentzen, H. Aanæs, and R. Larsen, *Shape analysis using the auto diffusion function*, Computer Graphics Forum **28** (2009), no. 5, 1405–1413.
34. N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann, *Robust global registration*, Proc. SGP, 2005.
35. P. Glomb, *Detection of interest points on 3D data: Extending the harris operator*, Computer Recognition Systems 3, Advances in Soft Computing, vol. 57, Springer Berlin / Heidelberg, May 2009, pp. 103–111.
36. C. Harris and M. Stephens, *A combined corner and edge detection*, Proc. Fourth Alvey Vision Conference, 1988, pp. 147–151.
37. L. Itti, C. Koch, and E. Niebur, *A model of saliency-based visual attention for rapid scene analysis*, Trans. PAMI **20** (1998), no. 11.
38. A. Johnson, *Spin-images: A representation for 3-d surface matching*, Ph.D. thesis, Carnegie Mellon University, 2007.
39. A. E. Johnson and M. Hebert, *Using spin images for efficient object recognition in cluttered 3D scenes*, Trans. PAMI **21** (1999), no. 5, 433–449.
40. M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, *Symmetry descriptors and 3D shape matching*, Proc. SGP, 2004, pp. 115–123.
41. V. G. Kim, Y. Lipman, and T. Funkhouser, *Blended intrinsic maps*, Proc. SIGGRAPH, 2011.
42. P. Koehl, *Protein structure similarities*, Current Opinion in Structural Biology **11** (2001), no. 3, 348–353.
43. J. J. Koenderink and A. J. van Doorn, *Surface shape and curvature scales*, Image and vision computing **10** (1992), no. 8, 557–564.
44. I. Kokkinos and A. Yuille, *Scale Invariance without Scale Selection*, Proc. CVPR, 2008.
45. M. Kolomenkin, I. Shimshoni, and A. Tal, *On edge detection on surfaces,*, Proc. CVPR, 2009.
46. M. Körtgen, G.J. Park, M. Novotni, and R. Klein, *3d shape matching with 3d shape contexts*, Central European Seminar on Computer Graphics, 2003.
47. A. Kovnatsky, M. M. Bronstein, A. M. Bronstein, and R. Kimmel, *Photometric heat kernel signatures*, Proc. Conf. on Scale Space and Variational Methods in Computer Vision (SSVM), 2011.
48. B. Lévy, *Laplace-Beltrami eigenfunctions towards an algorithm that "understands" geometry*, Int'l Conf. Shape Modeling and Applications, 2006, The use of Laplace-Belrami operator for shape analysis and synthesis.
49. R. Litman, A. M. Bronstein, and M. M. Bronstein, *Diffusion-geometric maximally stable component detection in deformable shapes*, Computers and Graphics (2011).
50. T.W.R. Lo and J.P. Siebert, *Local feature extraction and matching on range images: 2.5 D SIFT*, Computer Vision and Image Understanding **113** (2009), no. 12, 1235–1250.
51. D. Lowe, *Distinctive image features from scale-invariant keypoint*, IJCV (2004).
52. E. Magid, O. Soldea, and E. Rivlin, *A comparison of gaussian and mean curvature estimation methods on triangular meshes of range image data*, Computer Vision and Image Understanding **107** (2007), no. 3, 139–159.
53. M. Mahmoudi and G. Sapiro, *Three-dimensional point cloud recognition via distributions of geometric distances*, Graphical Models **71** (2009), no. 1, 22–31.
54. S. Manay, B.W. Hong, A.J. Yezzi, and S. Soatto, *Integral invariant signatures*, Computer Vision-ECCV 2004 (2004), 87–99.

55. J. Matas, O. Chum, M. Urban, and T. Pajdla, *Robust wide-baseline stereo from maximally stable extremal regions*, Image and Vision Computing **22** (2004), no. 10, 761–767.
56. D. Mateus, R. P. Horaud, D. Knossow, F. Cuzzolin, and E. Boyer, *Articulated shape matching using laplacian eigenfunctions and unsupervised point registration*, Proc. CVPR (2008).
57. M. Meyer, M. Desbrun, P. Schroder, and A. H. Barr, *Discrete differential-geometry operators for triangulated 2-manifolds*, Visualization and Mathematics III (2003), 35–57, Cotangent weights discretization of the Laplace-Beltrami operator.
58. N. J. Mitra, L. J. Guibas, J. Giesen, and M. Pauly, *Probabilistic fingerprints for shapes*, Proc. SGP, 2006.
59. N. J. Mitra, L. J. Guibas, and M. Pauly, *Partial and approximate symmetry detection for 3D geometry*, ACM Trans. Graphics **25** (2006), no. 3, 560–568.
60. L. Najman and M. Couprie, *Building the component tree in quasi-linear time*, IEEE Trans. Image Proc. **15** (2006), no. 11, 3531–3539.
61. V. Natarajan, P. Koehl, Y. Wang, and B. Hamann, *Visual analysis of biomolecular surfaces*, Visualization in Medicine and Life Sciences, Mathematics and Visualization, 2008, pp. 237–255.
62. M. Ovsjanikov, J. Sun, and L. J. Guibas, *Global intrinsic symmetries of shapes*, Computer Graphics Forum, vol. 27, 2008, pp. 1341–1348.
63. M. Pauly, R. Keiser, and M. Gross, *Multi-scale feature extraction on point-sampled surfaces*, Computer Graphics Forum, vol. 22, 2003, pp. 281–289.
64. M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. J. Guibas, *Discovering structural regularity in 3D geometry.*
65. U. Pinkall and K. Polthier, *Computing discrete minimal surfaces and their conjugates*, Experimental mathematics **2** (1993), no. 1, 15–36, Cotangent weights discretization of the Laplace-Beltrami operator.
66. D. Raviv, A. M. Bronstein, M. M. Bronstein, and R. Kimmel, *Symmetries of non-rigid shapes*, Proc. NRTL, 2007.
67. D. Raviv, A. M. Bronstein, M. M. Bronstein, R. Kimmel, and G. Sapiro, *Diffusion symmetries of non-rigid shapes*, Proc. 3DPVT, 2010.
68. D. Raviv, M. M. Bronstein, A. M. Bronstein, and R. Kimmel, *Volumetric heat kernel signatures*, Proc. ACM Multimedia Workshop on 3D Object Retrieval, 2010.
69. M. Reuter, *Hierarchical shape segmentation and registration via topological features of laplace-beltrami eigenfunctions*, International Journal of Computer Vision **89** (2010), no. 2, 287–308.
70. M. Reuter, F.-E. Wolter, and N. Peinecke, *Laplace-spectra as fingerprints for shape matching*, Proc. ACM Symp. Solid and Physical Modeling, 2005, pp. 101–106.
71. R. M. Rustamov, *Laplace-Beltrami eigenfunctions for deformation invariant shape representation*, Proc. SGP, 2007, pp. 225–233.
72. O. Scherzer, *Handbook of mathematical methods in imaging*, Springer Verlag, 2010.
73. P. Shilane and T. Funkhauser, *Selecting distinctive 3D shape descriptors for similarity retrieval*, Proc. Shape Modelling and Applications, 2006.
74. I. Sipiran and B. Bustos, *A robust 3D interest points detector based on Harris operator*, 2010.
75. J. Sivic and A. Zisserman, *Video Google: A text retrieval approach to object matching in videos*, Proc. CVPR, 2003.
76. P. Skraba, M. Ovsjanikov, F. Chazal, and L. Guibas, *Persistence-based segmentation of deformable shapes*, Proc. NORDIA, 2010, pp. 45–52.
77. N. Sochen, R. Kimmel, and R. Malladi, *A general framework for low level vision*, IEEE Trans. Image Processing **7** (1998), no. 3, 310–318.
78. J. Starck and A. Hilton, *Correspondence labelling for widetimeframe free-form surface matching*, Proc. ICCV, 2007.
79. C. Strecha, A. M. Bronstein, M. M. Bronstein, and P. Fua, *LDAHash: Improved matching with smaller descriptors*, Technical report, EPFL, 2010.

80. R. W. Sumner and J. Popović, *Deformation transfer for triangle meshes*, Proc. Conf. Computer Graphics and Interactive Techniques, 2004, pp. 399–405.
81. J. Sun, M. Ovsjanikov, and L. Guibas, *A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion*, Computer Graphics Forum, vol. 28, 2009, pp. 1383–1392.
82. N. Thorstensen and R. Keriven, *Non-rigid shape matching using geometry and photometry*, Proc. CVPR, 2009.
83. R. Toldo, U. Castellani, and A. Fusiello, *Visual vocabulary signature for 3D object retrieval and partial matching*, Proc. 3DOR, 2009.
84. L. Torresani, V. Kolmogorov, and C. Rother, *Feature correspondence via graph matching: Models and global optimization*, Proc. ECCV, 2008, pp. 596–609.
85. C. Wang, M. M. Bronstein, A. M. Bronstein, and N. Paragios, *Discrete minimum distortion correspondence problems for non-rigid shape matching*, Proc. Conf. on Scale Space and Variational Methods in Computer Vision (SSVM), 2011.
86. M. Wardetzky, S. Mathur, F. Kälberer, and E. Grinspun, *Discrete Laplace operators: no free lunch*, Conf. Computer Graphics and Interactive Techniques, 2008, Analysis of different discretizations of the Laplace-Beltrami operator.
87. A. Zaharescu, E. Boyer, K. Varanasi, and R Horaud, *Surface feature detection and description with applications to mesh matching*, Proc. CVPR, 2009.