Chapter 8
Stable Semi-local Features for Non-rigid Shapes

Roee Litman, Alexander M. Bronstein, and Michael M. Bronstein

Abstract Feature-based analysis is becoming a very popular approach for geometric shape analysis. Following the success of this approach in image analysis, there is a growing interest in finding analogous methods in the 3D world. Maximally stable component detection is a low computation cost and high repeatability method for feature detection in images.In this study, a diffusion-geometry based framework for stable component detection is presented, which can be used for geometric feature detection in deformable shapes.

The vast majority of studies of deformable 3D shapes models them as the two-dimensional boundary of the volume of the shape. Recent works have shown that a volumetric shape model is advantageous in numerous ways as it better captures the natural behavior of non-rigid deformations. We show that our framework easily adapts to this volumetric approach, and even demonstrates superior performance.

A quantitative evaluation of our methods on the SHREC'10 and SHREC'11 feature detection benchmarks as well as qualitative tests on the SCAPE dataset show its potential as a source of high-quality features. Examples demonstrating the drawbacks of surface stable components and the advantage of their volumetric counterparts are also presented.

R. Litman (✉) · A.M. Bronstein
School of Electrical Engineering, Tel Aviv University, Tel Aviv, Israel
e-mail: roeelitm@post.tau.ac.il; alex@eng.tau.ac.il

M.M. Bronstein
Institute of Computational Science, Faculty of Informatics, Universita della Svizzera Italiana, Lugano, Switzerland
e-mail: michael.bronstein@usi.ch

8.1   Introduction

Following their success in image analysis, many feature-based methods have found their way into the world of 3D shape analysis. Feature descriptors play a major role in many applications of shape analysis, such as assembling fractured models [16] in computational archeology, or finding shape correspondence [36].

Some shape feature-works are inspired by and follow methods in image analysis, for example the histogram of intrinsic gradients used in [41] is similar in principle to the scale invariant feature transform (SIFT) [21] which has recently become extremely popular in image analysis. The concept of "bags of features" [32] was introduced as a way to construct global shape descriptors that can be efficiently used for large-scale shape retrieval [27, 37].

Other features were developed natively for 3D, as they rely on properties like the shape normal field, as in the popular spin image [17]. Another example is a family of methods based on the heat kernel [4, 35], describing the local heat propagation properties on a shape. These methods are deformation-invariant due to the fact that heat diffusion geometry is intrinsic, thus making descriptors based on it applicable in deformable shape analysis.

8.1.1   Related Work

The focus of this work is on another class of feature detection methods, one that finds stable components (or regions) in the analyzed image or shape. The origins of this approach are also in the image processing literature, this time in the form of the watershed transform [9, 39].

Matas et al. [22] introduced the stable component detection concept to the computer vision and image analysis community in the form of the maximally stable extremal regions (MSER) algorithm. This approach 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. This algorithm can be made very efficient [25] in certain settings, and was shown to have superior performance in a benchmark done by Mikolajczyk et al. [24]. A deeper inspection of the notion of region stability was done by Kimmel et al. [18], also proposing an alternative stability criteria. The MSER algorithm was also expanded to gray-scale volumetric images in [12], though this approach was tested only in a qualitative way, and not evaluated as a feature detector.

Methods similar to MSER have been explored in the works on topological persistence [13]. Persistence-based clustering [7] was used by Skraba et al. [33] to perform shape segmentation. More recently, Dey et al. [10] researched the persistence of the Heat Kernel Signature [35] for detecting features from partial

shape data. In Digne et al. [11], extended the notion of vertex-weighted component trees to meshes and proposed to detect MSER regions using the mean curvature. The approach was tested only in a qualitative way, and not evaluated as a feature detector.

A part of this study was published in the proceedings of the Shape Modeling International (SMI'11) conference [20].

8.1.2 Main Contribution

The contributions of our framework are three-fold:

First, in Sect. 8.3 we introduce a generic framework for stable component detection, which unifies vertex- and edge-weighted graph representations (as opposed to vertex-weighting used in image and shape maximally stable component detectors [11, 12, 22]). Our results (see Sect. 8.6) show that the edge-weighted formulation is more versatile and outperforms its vertex-weighted counterpart in terms of feature repeatability.

Second, in Sect. 8.2 we introduce diffusion geometric weighting functions suitable for both vertex- and edge-weighted component trees. We show that such functions are invariant under a large class of transformations, in particular, non-rigid inelastic deformations, making them especially attractive in non-rigid shape analysis. We also show several ways of constructing scale-invariant weighting functions. In addition, following Raviv et al. [29], we show that the suggested framework performs better on volumetric data removing the (sometimes) unwanted insensitivity to volume-changing transformations inherent to the surface model (see Fig. 8.8 for an illustration).

Third, in Sect. 8.6 we show a comprehensive evaluation of different settings of our method on a standard feature detection benchmark comprising shapes undergoing a variety of transformations. We also present a qualitative evaluation on the SCAPE dataset of scanned human bodies and demonstrate that our methods are capable of matching features across distinct data such as synthetic and scanned shapes.

8.2 Diffusion Geometry

Diffusion geometry is an umbrella term referring to geometric analysis of diffusion or random walk processes [8]. Let us consider the shape of a 3D physical object, modeled as a connected and compact region $X \subset \mathbb{R}^3$. The boundary of the region ∂X is a closed connected two-dimensional Riemannian manifold. In many application in graphics, geometry processing, and pattern recognition, one seeks geometric quantities that are invariant to inelastic deformations of the object X [19, 30, 35]. Traditionally in the computer graphics community, 3D shapes are modeled by

considering their 2D boundary surface ∂X, and deformations as isometries of ∂X preserving its Riemannian metric structure. In the following, we refer to such deformations as *boundary isometries*, as opposed to a smaller class of *volume isometries* preserving the metric structure inside the volume X (volume isometries are necessarily boundary isometries, but not vice versa – see Fig. 8.8 for an illustration). Raviv et al. [29] argued that the latter are more suitable for modeling realistic shape deformations that boundary isometries, which preserve the area of ∂X, but not necessarily the volume of X.

8.2.1 Diffusion on Surfaces

Recent line of works [4, 8, 19, 26, 30, 31, 35] studied intrinsic description of shapes by analyzing heat diffusion processes on ∂X, governed by the *heat equation*

$$\left(\frac{\partial}{\partial t} + \Delta_{\partial X} \right) u(t, x) = 0, \tag{8.1}$$

where $u(t, x) : [0, \infty) \times \partial X \rightarrow [0, \infty]$ is the heat value at a point x in time t, and $\Delta_{\partial X}$ is the positive-semidefinite Laplace-Beltrami operator associated with the Riemannian metric of ∂X. The solution of (8.1) is derived from

$$u(x, t) = \int_{\partial X} h_t(x, y) u_0(y) da(y) \tag{8.2}$$

and is unique given the initial condition $u(0, x) = u_0(x)$, and a boundary condition if the manifold ∂X has a boundary. The Green's function $h_t(x, y)$ is called the *heat kernel* and represents the amount of heat transferred on ∂X from x to y in time t due to the diffusion process. The heat kernel is the non-shift-invariant impulse response of (8.1), i.e $h_t(x, x_0)$ it is the solution to a point initial condition $u(0, x) = \delta(x, x_0)$. A probabilistic interpretation of the heat kernel $h_t(x, y)$ is the transition probability *density* of a random walk of length t from the point x to the point y. In particular, the diagonal of the heat kernel or the *auto-diffusivity function* $h_t(x, x)$ describes the amount of heat remaining at point x after time t. Its value is related to the Gaussian curvature $K(x)$ through

$$h_t(x, x) \approx \frac{1}{4\pi t} \left(1 + \frac{1}{6} K(x)t + \mathcal{O}(t^2) \right). \tag{8.3}$$

which describes the well-known fact that heat tends to diffuse slower at points with positive curvature, and faster at points with negative curvature. Due to this relation, the auto-diffusivity function was used by Sun et al. [35] as a local surface descriptor referred to as *heat kernel signature* (HKS). Being intrinsic, the HKS is invariant to boundary isometries of ∂X.

The heat kernel is easily computed using the spectral decomposition of the Laplace-Beltrami operator [19],

$$h_t(x, y) = \sum_{i \geq 0} e^{-\lambda_i t} \phi_i(x)\phi_i(y), \tag{8.4}$$

where $\phi_0 = \text{const}, \phi_1, \phi_2, \ldots$ and $\lambda_0 = 0 \leq \lambda_1 \leq \lambda_2 \ldots$ denote, respectively, the eigenfunctions and eigenvalues of $\Delta_{\partial X}$ operator satisfying $\Delta_{\partial X}\phi_i = \lambda_i \phi_i$.

The parameter t can be given the meaning of *scale*, and the family $\{h_t\}_t$ of heat kernels can be thought of as a scale-space of functions on ∂X. By integrating over all scales, a *scale-invariant* version of (8.4) is obtained,

$$c(x, y) = \int_0^\infty h_t(x, y)dt = \sum_{i \geq 0} \left(\int_0^\infty e^{-\lambda_i t} dt \right) \phi_i(x)\phi_i(y) \tag{8.5}$$

$$= \sum_{i \geq 1} \frac{1}{\lambda_i} \phi_i(x)\phi_i(y)$$

This kernel is referred to as the *commute-time kernel* and can be interpreted as the transition probability density of a random walk of any length. Similarly, $c(x, x)$ express the probability density of remaining at a point x after any time

It is worthwhile noting that both the heat kernel and the commute time kernels constitute a family of low pass filters. In Aubry et al. [2], argued that for some shape analysis tasks kernels acting as band-pass filters might be advantageous. The proposed "wave kernel signature" is related to the physical model of a quantum particle on a manifold described by the Schrödinger equation. The study of this alternative model is beyond the scope of this paper.

8.2.2 Volumetric Diffusion

Instead of considering diffusion processes on the boundary surface ∂X, Raviv et al. [29] considered diffusion *inside* the volume X, arising from the Euclidean volumetric heat equation with Neumann boundary conditions,

$$\left(\frac{\partial}{\partial t} + \Delta \right) U(t, x) = 0 \quad x \in \text{int}(X);$$

$$\langle \nabla U(t, x), n(x) \rangle = 0 \quad x \in \partial X. \tag{8.6}$$

Here, $U(t, x) : [0, \infty) \times \mathbb{R}^3 \to [0, \infty]$ is the volumetric heat distribution, Δ is the Euclidean positive-semidefinite Laplacian, and $n(x)$ is the normal to the surface ∂X at point x. The heat kernel of the volumetric heat equation (8.6) is given, similarly to (8.4) by

$$H_t(x, y) = \sum_{i \geq 0} e^{-\Lambda_i t} \Phi_i(x) \Phi_i(y), \tag{8.7}$$

where Φ_i and Λ_i are the eigenfunctions and eigenvalues of Δ satisfying $\Delta \Phi_i = \Lambda_i \phi_i$ and the boundary conditions $\langle \nabla \Phi_i(x), n(x) \rangle = 0$. A volumetric version of the commute-time kernel can be created in a similar manner by integration over all values of t, yielding $C(x, y) = \sum_{i \geq 1} \Lambda_i^{-3/2} \Phi_i(x) \Phi_i(y)$. The diagonal of the heat kernel $H_t(x, x)$ gives rise to the *volumetric HKS* (vHKS) descriptor [29], which is invariant to volume isometries of X.

8.2.3 Computational Aspects

Both the boundary of an object discretized as a mesh and the volume enclosed by it discretized on a regular Cartesian grid can be represented in the form of an undirected graph. In the former case, the vertices of the mesh form the vertex set V while the edges of the triangles constitute the edge set E. In the latter case, the vertices are the grid point belonging to the solid, and the edge set is constructed using the standard 6- or 26-neighbor connectivity of the grid (for points belonging to the boundary, some of the neighbors do not exist). With some abuse of notation, we will denote the graph by $X = (V, E)$ treating, whenever possible, both cases in the same way. Due to the possibility to express all the diffusion-geometric constructions in the spectral domain, their practical computation boils down to the ability to discretize the Laplacian.

8.2.3.1 Surface Laplace-Beltrami Operator

In the case of 2D surfaces, the discretization of the Laplace-Beltrami operator of the surface ∂X and can be written in the generic matrix-vector form as $\Delta_{\partial X} \mathbf{f} = \mathbf{A}^{-1} \mathbf{W} \mathbf{f}$, where $\mathbf{f} = (f(v_i))$ is a vector of values of a scalar function $f : \partial X \to \mathbb{R}$ sampled on $V = \{v_1, \dots, v_N\} \subset \partial X$, $\mathbf{W} = \text{diag}\left(\sum_{l \neq i} w_{il}\right) - (w_{ij})$ is a zero-mean $N \times N$ matrix of weights, and $\mathbf{A} = \text{diag}(a_i)$ is a diagonal matrix of normalization coefficients [14, 40]. Very popular in computer graphics is the *cotangent weight* scheme [23, 28], where

$$w_{ij} = \begin{cases} (\cot \alpha_{ij} + \cot \beta_{ij})/2 & (v_i, v_j) \in E; \\ 0 & else, \end{cases} \tag{8.8}$$

where α_{ij} and β_{ij} are the two angles opposite to the edge between vertices v_i and v_j in the two triangles sharing the edge, and a_i are the discrete area elements. The eigenfunctions and eigenvalues of $\Delta_{\partial X}$ are found by solving the generalized eigendecomposition problem $\mathbf{W} \phi_i = \mathbf{A} \phi_i \lambda_i$ [19]. Heat kernels are approximated by taking a finite number of eigenpairs in the spectral expansion.

8.2.3.2 Volumetric Laplacian

In the 3D case, we used a ray shooting method to create rasterized volumetric shapes, i.e. every shape is represented as arrays of voxels on a regular Cartesian grid, allowing us to use the standard Euclidean Laplacian. The Laplacian was discretized using a 6-neighborhood stencil. We use the finite difference scheme to evaluate the second derivative in each direction in the volume, and enforced boundary conditions by zeroing the derivative outside the shape.

The construction of the Laplacian matrix under these conditions boils down to this element-wise formula (up to multiplicative factor):

$$(\Delta)_{ij} = \begin{cases} -1 & \text{if } i \neq j \text{ and } (v_i, v_j) \in E \\ -\sum_{k \neq j} (\Delta)_{kj} & \text{if } i = j \end{cases}$$

8.3   Maximally Stable Components

Let us now we go over some preliminary graph-theory terms, needed to cover the topic of the component-tree. As mentioned, we treat the discretization of a shape as an undirected graph $X = (V, E)$ with the vertex set V and edge set E. Two vertices v_1 and v_2 are said to be *adjacent* if $(v_1, v_2) \in E$. A *path* is an ordered sequence of vertices $\pi = \{v_1, \dots, v_k\}$ such that for any $i = 1, \dots, k-1$, v_i is adjacent to v_{i+1}. In this case, every pair of vertices on π are *linked* in X. If every pair of vertices in X is linked, the graph is said to be *connected*. A *subgraph* of X is every graph for which $Y = (V' \subseteq V, E' \subseteq E)$, and will be denoted by $Y \subseteq X$. Such Y will be a (connected) *component* of X if it is a maximal connected subgraph of X (i.e. for any connected subgraph Z, $Y \subseteq Z \subseteq X$ implies $Y = Z$). A subset of the graph edges $E' \subseteq E$, induces the graph $Y = (V', E')$ where $V' = \{v \in V : \exists v' \in V, (v, v') \in E'\}$, i.e. V' is a vertex set is made of all vertices belonging to an edge in E'.

A component tree can be built only on a weighted graph. A graph is called *vertex-weighted* if it is equipped with a scalar function $f : V \to \mathbb{R}$. Similarly, an *edge-weighted* graph is one that is equipped with a function $d : E \to \mathbb{R}$ defined on the edge set. In what follows, we will assume both types of weights to be non-negative.

In order to define the MSER algorithm on images, some regular connectivity (e.g., four-neighbor) was used. Gray-scale images may be represented as vertex-weighted graphs where the intensity of the pixels is used as weights. Using a function measuring dissimilarity of pairs of adjacent pixels one can obtain edge weights, as done by Forssen in [15]. Edge weighting is more general than vertex weighting, which is limited to scalar (gray-scale) images.

8.3.1 Component Trees

The ℓ-*cross-section* of a graph X is the sub-graph created by using only weights smaller or equal to ℓ (assuming $\ell \geq 0$). If the graph has a vertex-weight $f : V \to \mathbb{R}$, its ℓ-cross-section is the graph induced by $E_\ell = \{(v_1, v_2) \in E : f(v_1), f(v_2) \leq \ell\}$. Similarly, for a graph with an edge-weight $d : E \to \mathbb{R}$ the ℓ-cross-section is induced by the edge subset $E_\ell = \{e \in E : d(e) \leq \ell\}$. A connected component of the ℓ-cross-section is called the ℓ-*level-set* of the weighted graph.

The *altitude* of a component C, denoted by $\ell(C)$, is defined as the minimal ℓ for which C is a component of the ℓ-cross-section. Altitudes establish a partial order relation on the connected components of X as any component C is contained in a component with higher altitude.

The set of all such pairs $(\ell(C), C)$ therefore forms a tree called the *component tree*. The component tree is a data structure containing the nesting hierarchy the level-sets of a weighted graph. Note that the above definitions are valid for both vertex- and edge-weighted graphs.

8.3.2 Maximally Stable Components

Since we represent a discretized smooth manifold (or a compact volume) by an undirected graph, a measure of *area* (or *volume*) can associate with every subset of the vertex set, and therefore also with every component. Even though we are dealing with both surface and volume, w.l.o.g we will refer henceforth only to surface area as a measure of a component C, which will be denoted by $A(C)$. When dealing with regular sampling, like in the case images, the area of C can be thought of as its cardinality (i.e. the number of pixels in it). In the case of non-regular sampling a better discretization is needed, and a discrete area element $da(v)$ is associated with each vertex v in the graph. The area of a component in this case is defined as $A(C) = \sum_{v \in C} da(v)$.

The process of detection is done on a sequence of nested components $\{(\ell, C_\ell)\}$ forming a branch in the component tree. We define the *stability* of C_ℓ as a derivative along the latter branch:

$$s(\ell) = \frac{\text{Change in altitude}}{\text{\% Change in area}} = \frac{A(C_\ell)}{\frac{d}{d\ell} A(C_\ell)}. \tag{8.9}$$

In other words, the less the area of a component changes with the change of ℓ, the more stable it is. A component C_{ℓ^*} is called *maximally stable* if the stability function has a local maximum at ℓ^*. As mentioned before, maximally stable components are widely known in the computer vision literature under the name of *maximally stable extremal regions* or *MSERs* for short [22], with $s(\ell^*)$ usually referred to as the region *score*.

Pay attention that while in their original definition, both MSERs and their volumetric counterpart [12] were defined on a component tree of a vertex-weighted graph, while the latter definition allows for edge-weighted graphs as well and therefore more general. The importance of such an extension will become evident in the sequel. Also, the original MSER algorithm [22] assumes the vertex weights to be quantized, while our formulation is suitable for scalar fields whose dynamic range is unknown a priori (this has some disadvantages, as will be seen up next).

8.3.3 Computational Aspects

Najman et al. [25] introduced quasi-linear time algorithm for the construction of vertex-weighted component trees, and its adaptation to the edge-weighted case is quite straightforward. The algorithm is based on the observation that the vertex set V can be partitioned into disjoint sets which are merged together as one goes up in the tree. Maintaining and updating such a partition can be performed very efficiently using the *union-find* algorithm and related data structures. The resulting tree construction complexity is $O(N \log \log N)$. However, since the weights must be sorted prior to construction, the latter complexity is true only if the weights are quantized over a known range (which is not the case in weights we used), otherwise the complexity becomes $O(N \log N)$.

The stability function (8.9) contains a derivative along a branch of the component tree. Given a branch of the tree is a sequence of nested components $C_{\ell_1} \subseteq C_{\ell_2} \subseteq \cdots \subseteq C_{\ell_K}$, the derivative was approximated using finite differences scheme:

$$s(\ell_k) \approx \frac{A(C_{\ell_k})}{A(C_{\ell_{k+1}}) - A(C_{\ell_{k-1}})}(\ell_{k+1} - \ell_{k-1}). \tag{8.10}$$

Starting from the leaf nodes, a single pass over every branch of the component tree is in order to evaluate the function and detected its local maxima. Next, we filter out maxima with too low values of s (more details about this in the detector result section). Finally, we keep only the bigger of two nested maximally stable regions if they are overlapping by more that a predefined threshold.

8.4 Weighting Functions

Unlike images where pixel intensities are a natural vertex weight, 3D shapes generally do not have any such field. This could be a possible reason why MSER was shown to be extremely successful as a feature detector in images, but equivalent techniques for 3D shapes are quite rare. This method was recently proposed in [11], but due to the fact it uses mean curvature it is not deformation invariant and therefore not suitable for the analysis deformable shape. The diffusion geometric framework used in [33] is more appropriate for this task, as it allows a robust way

to analyze deformable shapes. We follow this approach and show that it allows the construction both vertex and edge weights suitable for the definition of maximally stable components with many useful properties.

Note that even though all of the following weighting functions are defined on the surface of the shape (usually by using $h_t(x, y)$), they are easily adapted to the volumetric model (usually by using $H_t(x, y)$ instead).

The discrete auto-diffusivity function is a trivial vertex-weight when using diffusion geometry, and can be directly used given its value on a vertex v:

$$f(v) = h_t(v, v). \tag{8.11}$$

As will be with all diffusion geometric weights, the latter weights are intrinsic. Therefore, maximally stable components defined this way are invariant to non-rigid bending. In images, weighting functions are based on intensity values and therefore contain all the data needed abut the image. The above weighting function, however, does not capture the complete intrinsic geometry of the shape, and depends on the scale parameter t.

Every scalar field may also be used as an edge-weight simply by using $d(v_1, v_2) = |f(v_1) - f(v_2)|$. As mentioned, edge weights are a more flexible and allow us more freedom in selecting how to incorporate the geometric information. For example, a vector-valued field defined on the vertices of the graph can be used to define an edge weighting scheme by weighting an edge by the distance between its vertices' values, as done in [15].

There are ways to define edge weights without the usage of vector field defined on the vertices. Lets take the discrete heat kernel $h_t(v_1, v_2)$ as an example. Taking a weight function inversely proportional to its value is metric-like in essence, since it represents random-walk probability between v_1 and v_2. The resulting edge weight will be

$$d(v_1, v_2) = \frac{1}{h_t(v_1, v_2)}. \tag{8.12}$$

This function also contains fuller information about the shape's intrinsic geometry, for small values of t.

8.4.1 Scale Invariance

All of the three latter weighting functions are based on the heat kernel, and therefore are not scale invariant. If we globally scale the shape by a factor $a > 0$, both the time parameter and the kernel itself will be scaled by a^2, i.e. the scaled heat kernel will be $a^2 h_{a^2 t}(v_1, v_2)$. The volumetric heat kernel $H_t(v_1, v_2)$ will be scaled differently and will become $a^3 H_{a^2 t}(v_1, v_2)$ (this is why $C(v_1, v_2)$ is constructed using the eigenvalues with a power of $-3/2$).

The commute time kernel is scale invariant, and could be used as a replacement to the heat kernel. However, the numerical computation of the commute time kernel

is more difficult as its coefficients decay polynomially (8.5), very slow compared to $h_t(v_1, v_2)$ whose coefficients decay exponentially (8.4). The slower decay translates to the need of many more eigenfunctions of the Laplacian for $c(v_1, v_2)$ to achieve the same accuracy as $h_t(v_1, v_2)$.

Let us point out that there is some invariance to scale, originating in the way the detector operates over the component tree (this is also notable in the detector result section). As noted by Matas et al. [22], the MSER detector is invariant to any monotonic transformation on the weights (originally pixel-intensities), a fact that can be harnessed to gain scale invariance in our detector. In practice, it is sufficient for us to limit the effect scaling on the weights to a monotonous transformation instead of completely undoing its effect. Such a weighting function will not be scale invariant by itself, nor the stability function (8.9) computed on such a component tree. However, the local maxima of (8.9), namely the stable components, will remain unaffected. An alternative to this approach would be designing a more sophisticated scale-invariant stability function. We intend to explore both of these options in follow up studies.

8.5   Descriptors

Usually, when using a feature-based approach, feature detection is followed by attaching a descriptor to each of the features. Once descriptors are manifested, we can measure similarity between a pair of features, which in turn, enables us to perform tasks like matching and retrieval. Since our detected feature are components of the shape, we first create a point-wise descriptor of the form $\alpha : V \to \mathbb{R}^q$ and then aggregate al the point-descriptors into a single region descriptor.

8.5.1   Point Descriptors

We consider a descriptor suitable for non-rigid shapes proposed in [35] – the *heat kernel signature* (HKS) HKS is computed by sampling the values of the discrete auto-diffusivity function at vertex v at multiple times, $\alpha(v) = (h_{t_1}(v, v), \ldots, h_{t_q}(v, v))$, where $t_1, \ldots, t_q$ are some fixed time values. The resulting descriptor is a vector of dimensionality q at each vertex. Since the heat kernel is an intrinsic quantity, the HKS is invariant to isometric transformations of the shape.

8.5.1.1   Scale-Invariant Heat Kernel Signature

The fact that HKS is based on $h_t(v, v)$, means it also inherits the drawback of its dependence on the scaling of the shape. As mentioned, scaling the shape globally by a factor $a > 0$ will result in the scaled heat kernel $a^2 h_{a^2 t}(v_1, v_2)$. A way of rendering $h_t(v, v)$ scale invariant was introduced in [4], by performing a sequence of

transformations on it. First, we sample the heat kernel with logarithmical spacing in time. Then, we take the logarithm of the samples and preform a numerical derivative (with respect to time of the heat kernel) to undo the multiplicative constant. Finally, we perform the discrete Fourier transform followed by taking the absolute value, to undo the scaling of the time variable. Note that the latter sequence of transformations will also work on $H_t(v, v)$, as the effect of scaling differs only in the power of the multiplicative constant.

This yields the *modified heat kernel* of the form

$$\hat{h}_\omega(v, v) = \left| \mathscr{F} \left\{ \frac{\partial \log h_t(v, v)}{\partial \log t} \right\} (\omega) \right| \tag{8.13}$$

where ω denotes the frequency variable of the Fourier transform. The scale-invariant version of the HKS descriptor (SI-HKS) is obtained by replacing h_t with $\hat{h}_\omega$, yielding $\alpha(v) = (\hat{h}_{\omega_1}(v, v), \ldots, \hat{h}_{\omega_q}(v, v))$, where $\omega_1, \ldots, \omega_q$ are some fixed frequency values.

8.5.1.2   Vocabulary Based Descriptors

Another method for the descriptor construction is following the *bag of features* paradigm [32]. Ovsjanikov et al. [27] used this approach to create a global shape descriptor using point-wise descriptors. In the bag of features approach we perform off-line clustering of the descriptor space, resulting in a fixed "geometric vocabulary" $\alpha_1, \ldots, \alpha_p$. We then take any point descriptor $\alpha(v)$, and represent it using the vocabulary by means of vector quantization. This results in a new point-wise p-dimensional vector $\theta(v)$, where each of its elements follow a distribution of the form $[\theta(v)]_l \propto e^{-\|\alpha(v)-\alpha_l\|^2/2\sigma^2}$. The vector is then normalized in such a way that the elements $\theta(v)$ sum to one. Setting $\sigma = 0$ is a special case named "hard vector quantization" where the descriptor boils down to a query result for the nearest neighbor of $\alpha(v)$ in the vocabulary. In other words, we will get $[\theta(v)]_l = 1$ for α_l being the closest vocabulary element to $\alpha(v)$ in the descriptor space, and zero elsewhere.

8.5.2   Region Descriptors

After creating a descriptor $\alpha(v)$ at each vertex $v \in V$, we need to gather the information from a subset of vertices, i.e. component $C \subset V$. This will result in a *region descriptor*. The simplest way to do this is by computing the average of α in C, weighted by the discrete area elements:

$$\beta(C) = \sum_{v \in C} \alpha(v) da(v). \tag{8.14}$$

The resulting region descriptor $\beta(C)$ is a vector of the same dimensionality q as the point descriptor α.

Other methods to create region descriptors from point descriptors such as the region covariance descriptors [38], are beyond the scope of this text. The latter method may provide higher discriminativity due to the incorporation of spatial contexts.

8.6 Results

8.6.1 Datasets

The proposed approaches were tested both qualitatively and quantitatively. All datasets mentioned below were given as triangular meshes, i.e. as 2D manifolds. In the following experiments, meshes were down-sampled to at most 10, 000 vertices. For the computation of the volumetric regions, meshes were rasterized in a cube with variable number of voxels per dimension (usually around 100–130) in order for the resulting shapes to contain approximately 45,000 voxels.

8.6.1.1 Data for Quantitative Tests

While almost every data set can be used for a qualitative evaluation, only datasets with additional ground-truth data can be used to quantify the performance of the algorithm. We chose two such data sets: the SHREC'10 [6] and SHREC'11 [3] feature detection and description benchmarks.

SHREC'10[6] was used for evaluation only of the 2D version of the algorithm. The dataset consisted of three shape classes (human, dog, and horse), with simulated transformations applied to them. Shapes are represented as triangular meshes with approximately 10,000–50,000 vertices.

Each shape class is given in a so-called "native" form, coined *null*, and also in nine categories of transformations: *isometry* (non-rigid almost inelastic deformations), *topology* (welding of shape vertices resulting in different triangulation), *micro holes* and *big holes* simulating missing data and occlusions, *global* and *local scaling*, *additive Gaussian noise*, *shot noise*, and *downsampling* (less than 20 % of the original points). All mentioned transformations appeared in five different strengths, and are combined with isometric deformations. The total number of transformations per shape class is 45 + 1 null shape, i.e. 138 shapes in total.

Vertex-wise correspondence between the transformed and the null shapes was given and used as the ground-truth in the evaluation of region detection repeatability. Since all shapes exhibit intrinsic bilateral symmetry, best results over the ground-truth correspondence and its symmetric counterpart were used.

SHREC'11 [3] was used for the comparison between the surface and volumetric approaches, due to the fact that results of both methods were too similar on SHREC'10. Having a wider and more challenging range and strength of transformations present in the SHREC'11 corpus was needed to emphasize the difference. The SHREC'11 dataset was constructed along the guidelines of its predecessor, SHREC'10. It contains one class of shapes (human) given in a null form, and also in 11 categories of transformations, in transformation appeared in five different strengths combined with isometric deformations. The strength of transformation is more challenging, in comparison to SHREC'10. The total number of shapes is 55 + 1 null shape.

Most of the transformations appear also in SHREC'10: *isometry*, *micro holes* and *big holes*, *global*, *additive Gaussian noise*, *shot noise*, and *downsampling*. Two transformations were discarded (*topology* and *local scaling*) and some new ones were introduced: *affine transformation*, *partial* (missing parts), *rasterization* (simulating non-pointwise topological artifacts due to occlusions in 3D geometry acquisition), and *view* (simulating missing parts due to 3D acquisition artifacts). Some of the mentioned transformations are not compatible with our volumetric rasterization method. We did not include in our experiments the following transformations: big holes, partial and view.

Vertex-wise correspondence were given like in SHREC'10, including bilateral symmetry. Volumetric ground-truth had to be synthesized, however. For that purpose, the surface voxels of two shapes were first matched using the groundtruth correspondences; then, the interior voxels were matched using an onion-peel procedure.

8.6.1.2 Data for Qualitative Tests

In addition to the already mentioned sets, three more datasets without groundtruth correspondence were used to demonstrate the performance of the proposed method visually:

TOSCA [5] The dataset contains 80 high-resolution nonrigid shapes in a variety of poses, including cat, dog, wolves, horse, 6 centaur, gorilla, female figures, and two different male figures. Shapes have a varying number vertices, usually about 50,000. This dataset gives a good example of the potential of the detected features to be used for partial-matching of shapes (see Figs. 8.1–8.3).

SCAPE[1] The dataset contains watertight scanned human figures, containing around 12.5 K vertices, in various poses. This dataset gives a hint on performance of the detector on real life (not synthetic) data. Figure 8.7 shows that the detected components are consistent and remain invariant under pose variations.

Deformation Transfer for Triangle Meshes dataset by Sumner et al.[34] contains a few shapes, out of which we chose two animated sequences of a horse shape represented as a triangular mesh with approximately 8.5 K vertices.

Fig. 8.1 Maximally stable regions detected on different shapes from the TOSCA dataset. Note the invariance of the regions to strong non-rigid deformations. Also observe the similarity of the regions detected on the female shape and the upper half of the centaur (compare to the male shape from Fig. 8.2). Regions were detected using $h_t(v, v)$ as vertex weight function, with $t = 2,048$

Fig. 8.2 Maximally stable regions detected on shapes from the SHREC'10 dataset using the vertex weight $h_t(v, v)$ with $t = 2,048$. *First row*: different approximate isometries of the human shape. *Second row*: different transformations (*left*-to-*right*): holes, localscale, noise, shotnoise and scale

One sequence includes a series of boundary and volume isometries (gallop), while the other includes series of non-volume-preserving boundary isometries (collapsing). Figure 8.8 shows that while the surface MSERs are invariant to both types of transformations, the proposed volumetric MSERs remain invariant only under volume isometries, changing quite dramatically if the volume is not preserved – a behavior consistent with the physical intuition.
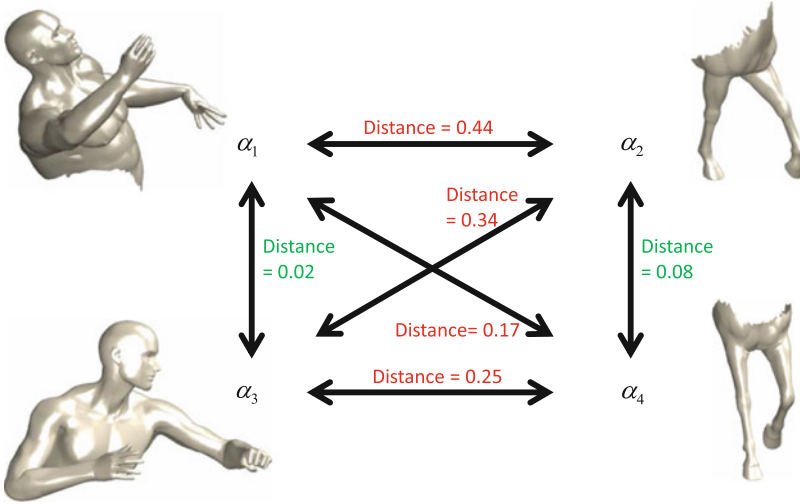
Fig. 8.3 A toy example showing the potential of the proposed method for partial shape matching. Shown are four maximally stable regions detected on the surface of three shapes from the TOSCA dataset using the edge weight $1/h_t(v_1, v_2)$ with $t = 2,048$. The *top* two regions are the torso and the front legs of a centaur (marked with α_1 and α_2 respectively), and on the *bottom* are the torso of a human and the front legs of a horse (marked with α_3 and α_4 respectively). Each of the regions is equipped with a SI-HKS descriptor, and L_2 distance is shown between every pair

8.6.2 Detector Repeatability

8.6.2.1 Evaluation Methodology

We follow the spirit of Mikolajczyk et al. [24] in the evaluation of the proposed feature detector (and also of the descriptor, later on). The performance of a feature detector is measured mainly by its *repeatability*, defined as the percentage of regions that have corresponding counterpart in a transformed version of the shape. In order to measure this quantity we need a rule to decide if two regions are "corresponding". Every comparison is done between a transformed shape and its original null shape, coined Y and X respectively. We will denote the regions detected in X and Y as $X_1, \ldots, X_m$ and $Y_1, \ldots, Y_n$. In order to perform a comparison, we use the ground-truth correspondence to project a region Y_j onto X, and will denote it's projected version as X'_j. We define the *overlap* of two regions X_i and Y_j as the following area ratio:

$$O(X_i, X'_j) = \frac{A(X_i \cap X'_j)}{A(X_i \cup X'_j)} = \frac{A(X_i \cap X'_j)}{A(X_i) + A(X'_j) - A(X_i \cap X'_j)} \qquad (8.15)$$

Fig. 8.4 Maximally stable regions detected on shapes from the SHREC'10 dataset using the edge weight $1/h_t(v_1, v_2)$ with $t = 2,048$. Region coloring is arbitrary

The *repeatability at overlap o* is defined as the percentage of regions in Y that have corresponding counterparts in X with overlap greater than o. An ideal detector has the repeatability of 100 % even for $o \rightarrow 1$. Note that comparison was defined single-sidedly due to the fact that some of the transformed shapes had missing data compared to the null shape. Therefore, unmatched regions of the null shape did not decrease the repeatability score, while regions in the transformed shape that had no corresponding regions in the null counterpart incurred a penalty on the score.

8.6.2.2 Surface Detector

Two vertex weight functions were compared: discrete heat kernel (8.11) with $t = 2,048$ and the commute time kernel. These two scalar fields were also used to construct edge weights according to $d(v_1, v_2) = |f(v_1) - f(v_2)|$. In addition, we used the fact that these kernels are functions of a pair of vertices to define edge weights according to (8.12). Unless mentioned otherwise, $t = 2,048$ was used for the heat kernel, as this setting turned out to give best performance on the SHREC'10 dataset.

We start by presenting a qualitative evaluation on the SHREC'10 and the TOSCA datasets. Regions detected using the vertex weight $h_t(v, v)$ with $t = 2,048$ are shown for TOSCA in Fig. 8.1 and for SHREC'10 in Fig. 8.2. Figure 8.4 depicts the maximally stable components detected with the edge weighting function $1/h_t(v_1, v_2)$ on several shapes from the SHREC'10 dataset. These results show robustness and repeatability of the detected regions under transformation. Surprisingly, many of these regions have a clear semantic interpretation, like limbs or head. In Addition, the potential of the proposed feature detector for partial shape matching can be seen by the similarly looking regions that are detected on the male and female shapes, and the upper half of the centaur (see a toy example in Fig. 8.3).

Ideally, we would like a detector to have perfect repeatability, i.e. to produce a large quantity of regions with a corresponding counterpart in the original shape. This is unfeasible, and all detectors will produce some poorly repeatable regions. However, if the repeatability of the detected regions is highly correlated with their
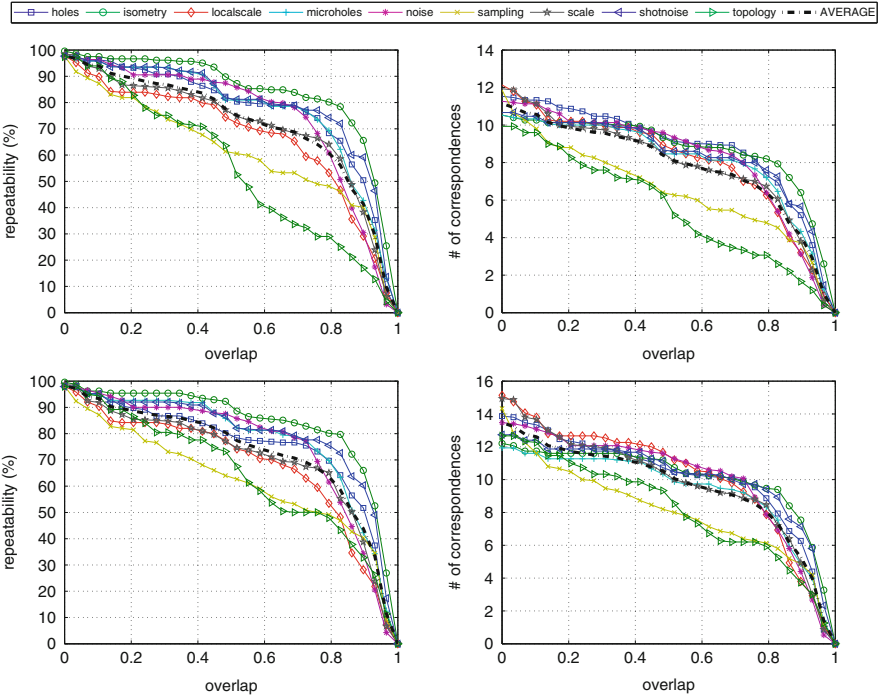
Fig. 8.5 Repeatability of maximally stable components with the vertex weight $h_t(v, v)$ (*first row*) and edge weight $1/h_t(v_1, v_2)$ (*second row*), $t = 2,048$

stability scores, a poor detector can still be deemed good by selecting a cutoff threshold on the stability score. In other words, set a minimum region stability value that is accepted by the detector, such that the rejected regions are likely not to be repeatable. This cutoff value is estimated based on the empirical 2D distributions of detected regions as a function of the stability score and the overlap with the corresponding ground-truth regions. Of course, this can only be done given ground-truth correspondences. In some of the tested detectors, a threshold was selected for stability score to minimize the detection of low-overlap regions, in order to give an estimate for the theoretical limits on the performance of the weighting functions.

We now show the performance of the best four weighting functions in Figs. 8.5 and 8.6. These figures depict the repeatability and the number of correctly matching regions as the function of the overlap. One can infer that scale-dependent weighting generally outperform their scale-invariant counterparts in terms of repeatability. This could be explained by the fact that we have selected the best time value for our dataset's common scale, whereas scale-invariant methods suffer from its larger degree of freedom. The scalar fields corresponding to the auto-diffusivity functions perform well both when used as vertex and edge weights. Best repeatability is achieved by the edge weighting function $1/h_t(v_1, v_2)$. Best scale invariant weighting by far is the edge weight $1/c(v_1, v_2)$.
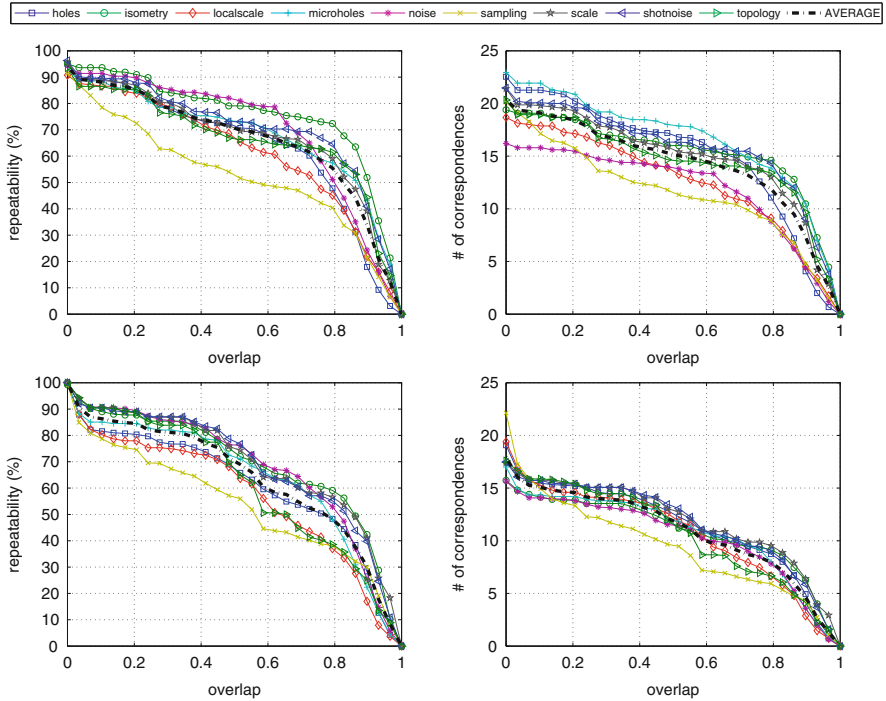
Fig. 8.6 Repeatability of maximally stable components with the edge weight $|h_t(v_1, v_1) - h_t(v_2, v_2)|$ (*first row*) and edge weight $1/c(v_1, v_2)$ (*second row*), $t = 2{,}048$

8.6.2.3 Volume Detector

In order to asses the differences between the 2D and the 3D of the approach, we performed two experiments comparing between 3D MSER and 2D MSER: comparison of invariance of the two methods to boundary and volume isometric deformations, and a quantitative comparison evaluating the sensitivity of two methods to shape transformations and artifacts on the SHREC'11 benchmark. In addition, we performed one evaluation of volumetric (3D) MSER invariance on scanned human figures.

As mentioned before, all the datasets used in our experiments were originally represented as triangular meshes and were rasterized and represented as arrays of voxels on a regular Cartesian grid.

In the first experiment, we applied the proposed approach to the SCAPE dataset [1], containing a scanned human figure in various poses. Figure 8.7 shows that the detected components are consistent and remain invariant under pose variations. In the second experiment, we used the data from Sumner et al. [34]. The dataset contained an animated sequence of a horse shape and includes a series of boundary and volume isometries (gallop) and series of non-volume-preserving

Fig. 8.7 Stable volumetric regions detected on the SCAPE data [1]. Shown are volumetric regions (*first* and *third columns*) and their projections onto the boundary surface (*second* and *fourth columns*). Corresponding regions are denoted with like colors. The detected components are invariant to isometric deformations of the volume
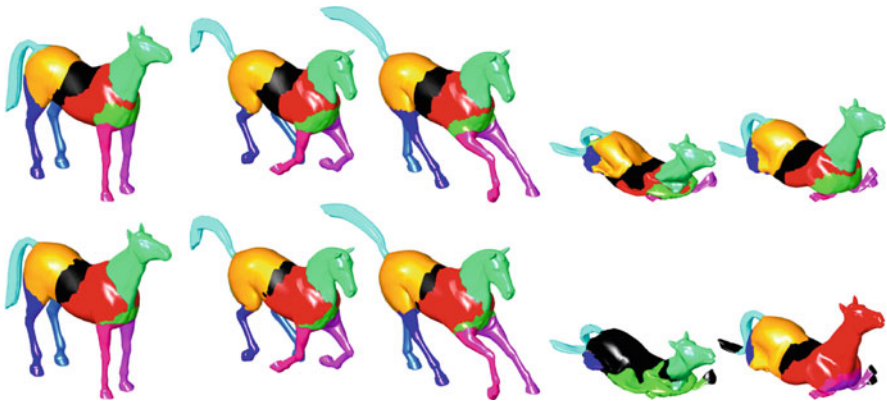

Fig. 8.8 Maximally stable components detected on two approximate volume isometries (*second* and *third columns*) and two volume-changing approximate boundary surface isometries (*two rightmost columns*) of the horse shape (*left column*). Stable regions detected on the boundary surface (2D MSER, *first row*) remain invariant to all deformations, while the proposed volumetric stable regions (3D MSER, *second row*) maintain invariance to the volume-preserving deformations only. This better captures natural properties of physical objects. Corresponding regions are denoted with like colors. For easiness of comparison, volumetric regions are projected onto the boundary surface

boundary isometries (collapsing). Figure 8.8 shows that while the surface MSERs are invariant to both types of transformations, the proposed volumetric MSERs remain invariant only under volume isometries, changing quite dramatically if the volume is not preserved – a behavior consistent with the physical intuition.
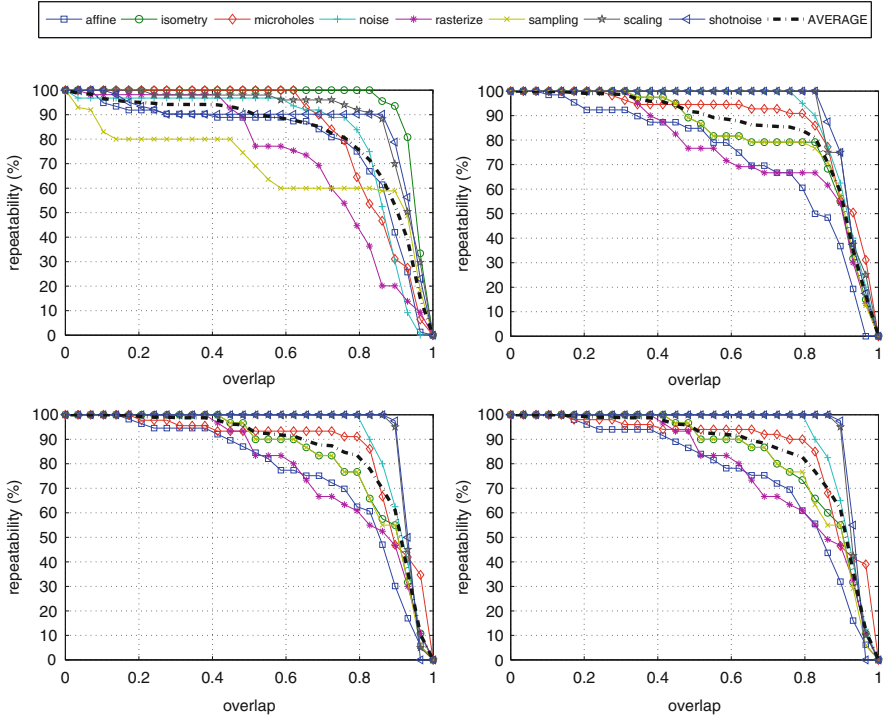
Fig. 8.9 Repeatability of region detectors on the SHREC'11 dataset. *Upper left*: 2D MSER using the edge weight $1/h_t(v_1, v_2)$, $t = 2,048$. *Upper right*: 3D MSER using the commute-time vertex-weight. *Lower left*: 3D MSER using the edge weight $1/H_t(v_1, v_2)$, $t = 2,048$. *Lower right*: 3D MSER using the vertex-weight $H_t(v, v)$, $t = 2,048$

In the third experiment, we used the SHREC'11 feature detection and description benchmark [3] to evaluate the performance of the 2D and 3D region detectors and descriptors under synthetic transformations of different type and strength.[1] As mentioned, some of the transformations in SHREC'11 are not compatible with our volumetric rasterization method, so we did not include in our experiments the big-holes, partial, and view transformations.

Figure 8.9 shows the repeatability of the 3D and 2D MSERs. We conclude that volumetric regions exhibit similar or slightly superior repeatability compared to boundary regions, especially for large overlaps (above 80 %). We attribute the slightly lower repeatability in the presence of articulation transformations ("isometry") to the fact that these transformations are almost exact isometries of the

---

[1]In this evaluation we used SHREC11, rather than SHREC10 that was used previously in 2D. this is due to the fact that results of the 3D and 2D versions were too similar on SHREC10, and dataset with a wider, and more challenging range and strength of transformations was needed to emphasize the difference.

boundary, while being only approximate isometries of the volume. Another reason for the latter degradation may be local topology changes that were manifested in the rasterization of the shapes in certain isometries. These topology changes appear only in the volumetric detector, and they affected the quality of detected regions in their vicinity. Although the construction of the MSER feature detector is not affine-invariant, excellent repeatability under affine transformation is observed. We believe that this and other invariance properties are related to the properties of the component trees (which are stronger than those of the weighting functions) and intend to investigate this phenomenon in future studies.

8.6.3 Descriptor Informativity

8.6.3.1 Evaluation Methodology

In these experiments, we aim to evaluate the informativity of region descriptors. This is done by measuring the relation between the overlap of two regions and their distance in the descriptor space.

Keeping the notation from the previous section, we set a minimum overlap $\rho = 0.75$ deeming two regions Y_i and X_j *matched*, i.e. if $o_{ij} = O(X_i', X_j) \geq \rho$ (X_i' is the projection of Y_i on the null shape X). This threshold is needed to constitute the matching ground-truth.

Given a region descriptor β on each of the regions, we set a threshold τ on the distance between the descriptors of Y_i and X_j in order to classify them as *positives*, i.e. if $d_{ij} = \|\beta(Y_i) - \beta(X_j)\| \leq \tau$. For simplicity we assume the distance between the descriptors to be the standard Euclidean distance. We define the *true positive rate* (TPR) and *false positive rate* (FPR) as the ratios

$$\text{TPR} = \frac{|\{d_{ij} \leq \tau\}|}{|\{o_{ij} \geq \rho\}|} \quad ; \quad \text{FPR} = \frac{|\{d_{ij} > \tau\}|}{|\{o_{ij} < \rho\}|}$$

The *receiver operator characteristic* (ROC) curve is obtained a set of pairs (FPR, TPR), created by varying the threshold τ. The *false negative rate* defined as FNR $= 1 - $ TPR. The *equal error rate* (EER) is the point on the ROC curve for which the false positive and false negative rates coincide. EER is used as a scalar measure for the descriptor informativity, where ideal descriptors have EER $= 0$.

Another descriptor performance criterion is created by finding for each X_i its nearest neighbor in the descriptor space $Y_{j^*(i)}$, namely $j^*(i) = \arg\min_j d_{ij}$. We then define the *matching score*, as the ratio of correct first matches for a given overlap ρ, and m, the total number of regions in X:

$$\text{score}(\rho) = \frac{|\{o_{ij^*(i)} \geq \rho\}|}{m}. \tag{8.16}$$

Table 8.1 Equal error rate (EER) performance of different maximally stable component detectors and descriptors ($t = 2,048$ was used in all cases). p denotes the vocabulary size in the bag of features region descriptors

Weighting function	HKS avgerage	HKS BoF($p = 10$)	HKS BoF($p = 12$)	SI-HKS avgerage	SI-HKS BoF($p = 10$)	SI-HKS BoF($p = 12$)
$h_t(v, v)$	0.311	0.273	0.278	0.093	0.091	0.086
$1/h_t(v_1, v_2)$	0.304	0.275	0.281	0.104	0.093	0.090
$\|h_t(v_1, v_1) - h_t(v_2, v_2)\|$	0.213	0.212	0.222	0.085	0.091	0.094
$1/c(v_1, v_2)$	0.260	0.284	0.294	0.147	0.157	0.148

8.6.3.2 Surface Descriptor

Given the maximally stable components detected by a detector, region descriptors were calculated. We used two types of point descriptors: the heat kernel signature (HKS), and its scale invariant version SI-HKS. Each of these two point descriptors was also used as a basis to create vocabulary based descriptors. Region descriptors were created from every point descriptor using averaging (8.14). In the following experiments, the HKS was created based on the heat kernel signature $h_t(v, v)$, sampled at six time values $t = 16, 22.6, 32, 45.2, 64, 90.5, 128$. The SI-HKS was created by sampling the heat kernel time values $t = 2^1, 2^{1+1/16}, \ldots, 2^{25}$ and taking the first six discrete frequencies of the Fourier transform, repeating the settings of [4]. Bags of features were tested on the two descriptors with vocabulary sizes $p = 10$ and 12, trained based on the SHREC'10 and TOSCA datasets. This sums up to a total of 6 descriptors – two in the "raw" form and four vocabulary based descriptors.

The four best weighting functions (shown in Figs. 8.5 and 8.6) were also selected to test region detectors. The performance in terms of EER of all the combinations of latter weighting functions and the six region descriptors is shown in Table 8.1.

Figures 8.10 and 8.11 show the number of correct first matches and the matching score as a function of the overlap for the two "raw" descriptors and the two best weighting functions: the vertex weight $h_t(v, v)$ and the edge weight $1/h_t(v_1, v_2)$. Figure 8.12 depicts the ROC curves of all of the descriptors based on maximally stable components of the same two weighting functions.

We conclude that the SI-HKS descriptor consistently exhibits higher performance in both the "raw" form and when using vocabulary, though the latter perform slightly worse. On the other hand, the bag of feature setting seems to improve the HKS descriptor in comparison to its "raw" form, though never reaching the scores of SI-HKS. Surprisingly, SI-HKS consistently performs better, even in transformations not including scaling, as can be seen from Figs. 8.10–8.11.
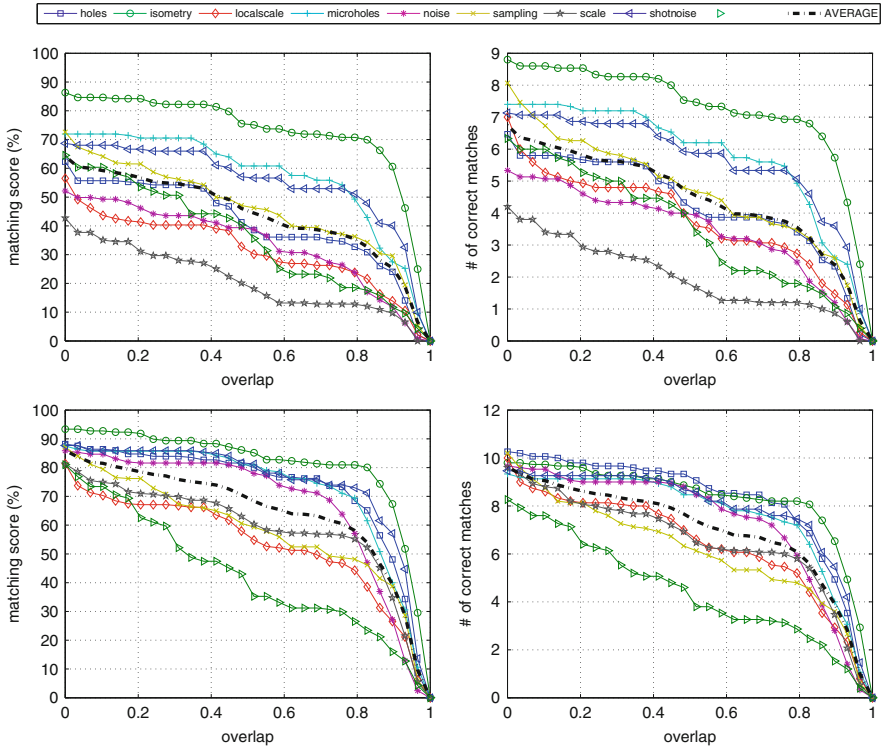
Fig. 8.10 Performance of region descriptors with regions detected using the vertex weight $h_t(v, v)$, $t = 2{,}048$. Shown are the HKS descriptor (*first row*) and SI-HKS descriptor (*second row*)

8.6.3.3   Volume Descriptor

Scale invariant volume HKS descriptor (SI-vHKS) were calculated for every volumetric stable region detected in the previous section. When testing the volumetric setting we used only the SI-vHKS descriptor due to its superior performance in 2D.

As with the surface descriptors, the matching score was measured for each of the volumetric regions and is shown in Fig. 8.13. For comparison, we used the SI-HKS descriptors on the boundary for the detected 2D regions. The combination of volumetric regions with volumetric descriptors exhibited highest performance over the entire range of deformations.

A region matching experiment was performed on the volumetric regions, seeking the nearest neighbors of a selected query region. The query region was taken from the SCAPE dataset, and the nearest neighbors were taken from the TOSCA dataset which differs considerably from the former (SCAPE shapes are 3D scans of human figures, while TOSCA contains synthetic shapes). Despite the mentioned difference, correct matches were found consistently, as can be seen in Fig. 8.14
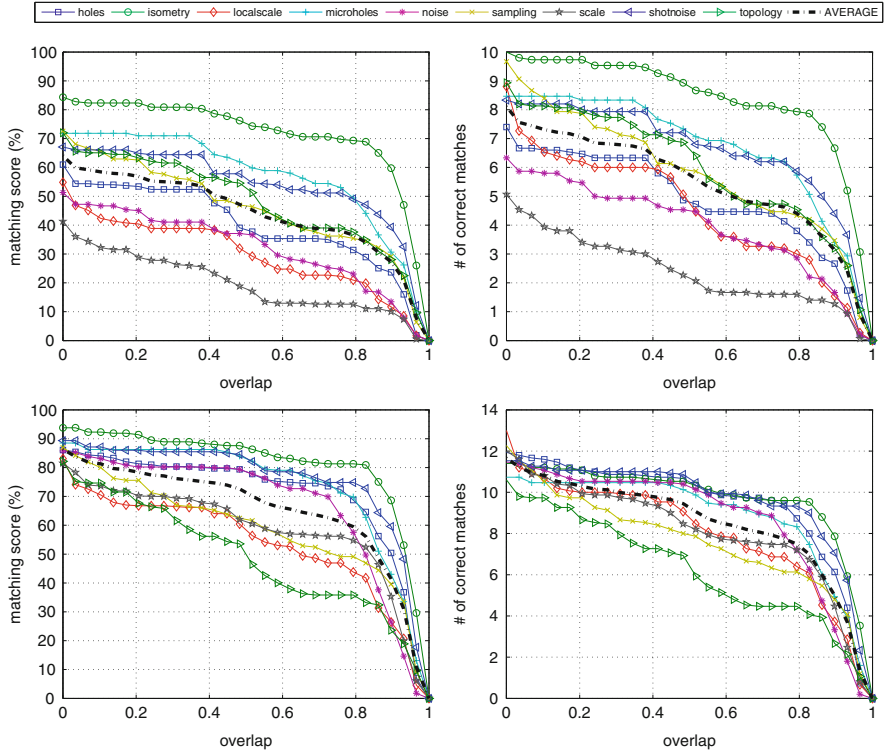
Fig. 8.11 Performance of region descriptors with regions detected using the edge weight $1/h_t(v_1, v_2)$, $t = 2,048$. Shown are the HKS descriptor (*first row*) and SI-HKS descriptor (*second row*)
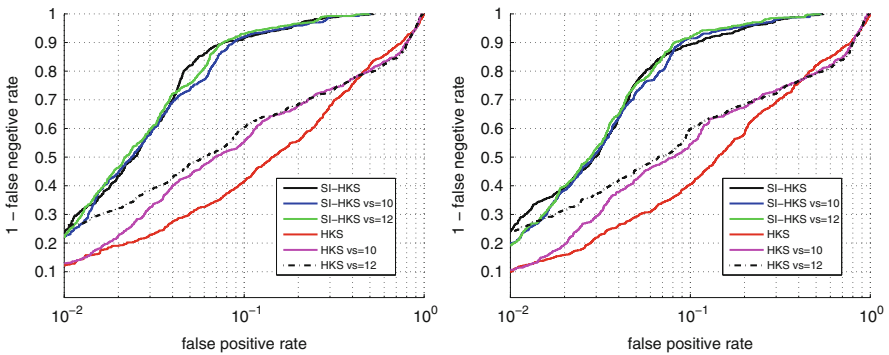

Fig. 8.12 ROC curves of different regions descriptors ("vs" stands for vocabulary size). The following detectors were used: vertex weight $h_t(v, v)$ (*left*), and edge weight $1/h_t(v_1, v_2)$ (*right*).
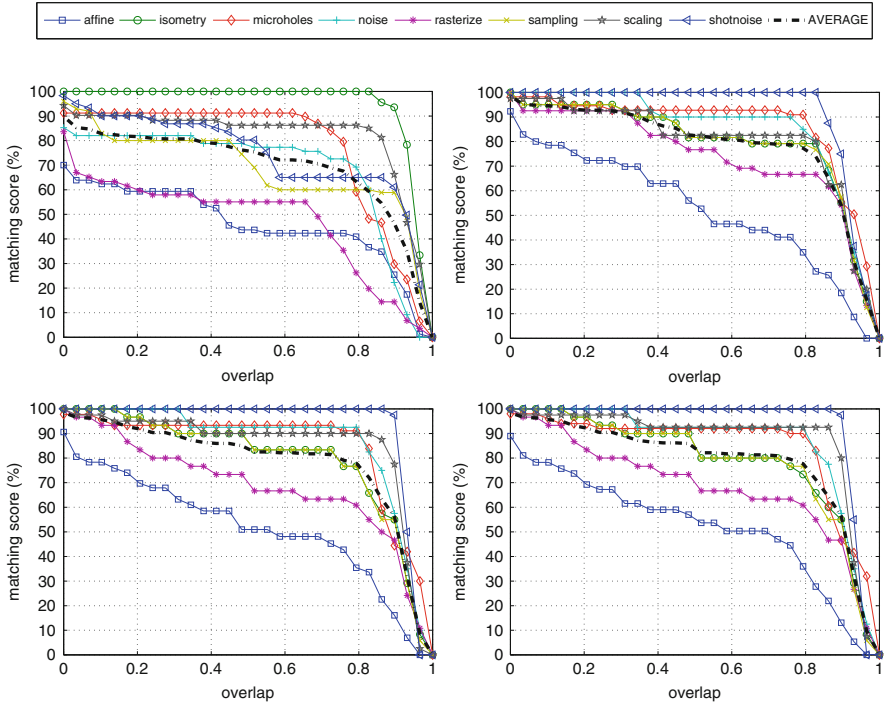
Fig. 8.13 Matching score of descriptors based on the 2D and 3D regions detected with the detectors shown in Fig. 8.9. Shown are the 2D SI-HKS (*upper left*) 3D SI-vHKS (*upper right* and two *bottom plots*) descriptors
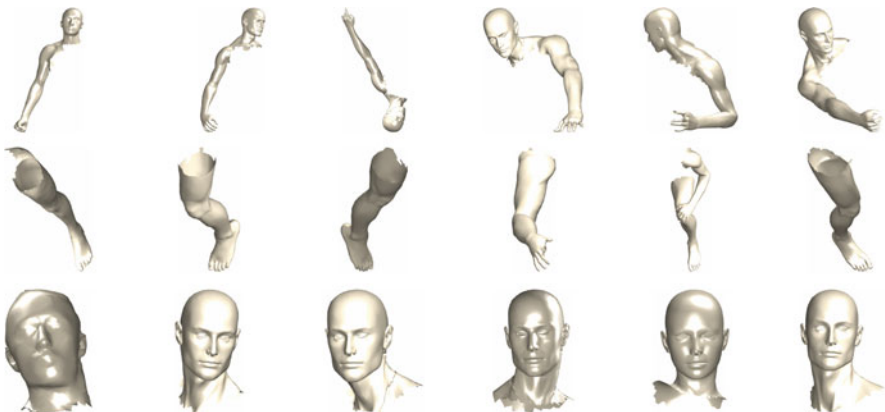

Fig. 8.14 Examples of closest matches found for different query regions from the SCAPE dataset on the TOSCA dataset. Shown from *left* to *right* are: query, 1st, 2nd, 4th, 10th, and 15th matches. Edge-weight $1/H_t(v_1, v_2)$ was used as the detector; average SI-vHKS was used as the descriptor

8.7   Conclusions

A generic framework for the detection of stable non-local features in deformable shapes is presented. This approach is based on a popular image analysis tool called MSER, where we maximize a stability criterion in a component tree representation of the shape. The framework is designed to unify the vertex or edge weights, unlike most of its predecessors. The use of diffusion geometry as the base of the weighting scheme make to detector invariant to non-rigid bending, global scaling and other shape transformations, a fact that makes this approach applicable in the challenging setting of deformable shape analysis.

The approach was shown to work with volumetric diffusion geometric analysis. In all experiments, our volumetric features exhibited higher robustness to deformation compared to similar features computed on the two-dimensional boundary of the shape. We also argue and exemplify that unlike features constructed from the boundary surface of the shape, our volumetric features are *not* invariant to volume-changing deformations of the solid object. We believe that this is the desired behavior in many applications, as volume isometries better model natural deformations of objects than boundary isometries.

We showed experimentally the high repeatability of the proposed features, which makes them a good candidate for a wide range of shape representation and retrieval tasks.

References

1. Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., Davis, J.: SCAPE: shape completion and animation of people. TOG **24**(3), 408–416 (2005)
2. Aubry, M., Schlickewei, U., Cremers, D.: The wave kernel signature-a quantum mechanical approach to shape analyis. In: Proceedings of the CVPR, Colorado Springs (2011)
3. Boyer, E., Bronstein, A.M., Bronstein, M.M., Bustos, B., Darom, T., Horaud, R., Hotz, I., Keller, Y., Keustermans, J., Kovnatsky, A., Litman, R., Reininghaus, J., Sipiran, I., Smeets, D., Suetens, P., Vandermeulen, D., Zaharescu, A., Zobel, V.: Shrec '11: Robust feature detection and description benchmark. In: Proceedings of the 3DOR, Llandudno, pp. 71–78 (2011)
4. Bronstein, M.M., Kokkinos, I.: Scale-invariant heat kernel signatures for non-rigid shape recognition. In: Computer Vision and Pattern Recognition, San Francisco, pp. 1704–1711 (2010)
5. Bronstein, A., Bronstein, M.M., Bronstein, M., Kimmel, R.: Numerical Geometry of Non-rigid Shapes. Springer, New York (2008)
6. Bronstein, A., Bronstein, M.M., Bustos, B., Castellani, U., Crisani, M., Falcidieno, B., Guibas, L.J., Kokkinos, I., Murino, V., Ovsjanikov, M., et al.: SHREC 2010: robust feature detection and description benchmark. In: Eurographics Workshop on 3D Object Retrieval (2010)

7. Chazal, F., Guibas, L.J., Oudot, S.Y., Skraba, P.: Persistence-based clustering in riemannian manifolds. Research Report RR-6968, INRIA (2009)
8. Coifman, R.R., Lafon, S.: Diffusion maps. Appl. Comput. Harmonic Anal. **21**(1), 5–30 (2006)
9. Couprie, M., Bertrand, G.: Topological grayscale watershed transformation. In: SPIE Vision Geometry V Proceedings, San Diego, vol. 3168, pp. 136–146 (1997)
10. Dey, T.K., Li, K., Luo, C., Ranjan, P., Safa, I., Wang, Y.: Persistent heat signature for pose-oblivious matching of incomplete models. Comput. Graph. Forum **29**(5), 1545–1554 (2010)
11. Digne, J., Morel, J.-M., Audfray, N., Mehdi-Souzani, C.: The level set tree on meshes. In: Proceedings of the Fifth International Symposium on 3D Data Processing, Visualization and Transmission, Paris (2010)
12. Donoser, M., Bischof, H.: 3d segmentation by maximally stable volumes (msvs). In: Proceedings of the 18th International Conference on Pattern Recognition, vol. 1, pp. 63–66. IEEE Computer Society, Los Alamitos (2006)
13. Edelsbrunner, H., Letscher, D., Zomorodian, A.: Topological persistence and simplification. Discret.Comput. Geom. **28**(4), 511–533 (2002)
14. Floater, M.S., Hormann, K.: Surface parameterization: a tutorial and survey. In: Advances in Multiresolution for Geometric Modelling, vol. 1, pp. 157–186. Springer, Berlin (2005)
15. Forssen, P.E.: Maximally stable colour regions for recognition and matching. In: Proceedings of the CVPR, Minneapolis, pp. 1–8 (2007)
16. Huang, Q.X., Flöry, S., Gelfand, N., Hofer, M., Pottmann, H.: Reassembling fractured objects by geometric matching. ACM Trans. Graph. **25**(3), 569–578 (2006)
17. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3D scenes. Trans. PAMI **21**(5), 433–449 (1999)
18. Kimmel, R., Zhang, C., Bronstein, A.M., Bronstein, M.M.: Are mser features really interesting? IEEE Trans. PAMI **33**(11), 2316–2320 (2011)
19. Levy, B.: Laplace-Beltrami eigenfunctions towards an algorithm that understands geometry. In: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006, pp. 13. IEEE Computer Society, Los Alamitos (2006)
20. Litman, R., Bronstein, A.M., Bronstein, M.M.: Diffusion-geometric maximally stable component detection in deformable shapes. Comput. Graph. **35**, 549–560 (2011)
21. Lowe, D.: Distinctive image features from scale-invariant keypoint. IJCV **60**(2), 91–110 (2004)
22. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image Vis. Comput. **22**(10), 761–767 (2004)
23. Meyer, M., Desbrun, M., Schroder, P., Barr, A.H.: Discrete differential-geometry operators for triangulated 2-manifolds. In: Visualization and Mathematics III, pp. 35–57. Springer, Berlin (2003)
24. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.V.: A comparison of affine region detectors. IJCV **65**(1), 43–72 (2005)
25. Najman, L., Couprie, M.: Building the component tree in quasi-linear time. IEEE Trans. Image Proc. **15**(11), 3531–3539 (2006)
26. Ovsjanikov, M., Sun, J., Guibas, L.: Global intrinsic symmetries of shapes. Comput. Graph. Forum **27**(5), 1341–1348 (2008)
27. Ovsjanikov, M., Bronstein, A.M., Bronstein, M.M., Guibas, L.J.: Shape google: a computer vision approach to isometry invariant shape retrieval. In: Computer Vision Workshops (ICCV Workshops), Kyoto, pp. 320–327 (2009)
28. Pinkall, U., Polthier, K.: Computing discrete minimal surfaces and their conjugates. Exp. Math. **2**(1), 15–36 (1993)
29. Raviv, D., Bronstein, M.M., Bronstein, A.M., Kimmel, R.: Volumetric heat kernel signatures. In: Proceedings of the ACM Workshop on 3D Object Retrieval, Firenze, pp. 39–44 (2010)
30. Reuter, M., Wolter, F.-E., Peinecke, N.: Laplace-spectra as fingerprints for shape matching. In: Proceedings of the ACM Symposium on Solid and Physical Modeling, Cambridge, pp. 101–106 (2005)
31. Rustamov, R.M.: Laplace-Beltrami eigenfunctions for deformation invariant shape representation. In: Proceedings of the SGP, Barcelona, pp. 225–233 (2007)

32. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: Proceedings of the CVPR, Madison, vol. 2, pp. 1470–1477 (2003)
33. Skraba, P. Ovsjanikov, M., Chazal, F., Guibas, L.: Persistence-based segmentation of deformable shapes. In: Proceedings of the NORDIA, San Francisco, pp. 45–52 (2010)
34. Sumner, R.W., Popović, J.: Deformation transfer for triangle meshes. ACM Transactions on Graphics (Proceedings of the SIGGRAPH), Los Angeles, vol. 23, pp. 399–405 (2004)
35. Sun, J., Ovsjanikov, M., Guibas, L.: A concise and provably informative multi-scale signature based on heat diffusion. Comput. Graph. Forum **28**, 1383–1392 (2009)
36. Thorstensen, N., Keriven, R.: Non-rigid shape matching using geometry and photometry. In: Computer Vision – ACCV 2009, Xi'an, vol. 5996, pp. 644–654 (2010)
37. Toldo, R., Castellani, U., Fusiello, A.: Visual vocabulary signature for 3d object retrieval and partial matching. In: Proceedings of the 3DOR, Munich, pp. 21–28 (2009)
38. Tuzel, O., Porikli, F., Meer, P.: Region covariance: a fast descriptor for detection and classification. In: Computer Vision ECCV 2006. Lecture Notes in Computer Science, vol. 3952, pp. 589–600. Springer, Berlin/Heidelberg (2006)
39. Vincent, L., Soille, P.: Watersheds in digital spaces: an efficient algorithm based on immersion simulations. IEEE Trans. PAMI, **13**(6), 583–598 (2002)
40. Wardetzky, M., Mathur, S., Kaelberer, F., Grinspun, E.: Discrete laplace operators: no free lunch. In: Proceedings of the of Eurographics Symposium on Geometry Processing, Barcelona, pp. 33–37 (2007)
41. Zaharescu, A., Boyer, E., Varanasi, K., Horaud, R.: Surface feature detection and description with applications to mesh matching. In: Proceedings of the CVPR, Miami, pp. 373–380 (2009)