

# Diffusion-geometric maximally stable component detection in deformable shapes

Roe Litman<sup>1</sup>, Alexander M. Bronstein<sup>1</sup>, Michael M. Bronstein<sup>2</sup>

## Abstract

Maximally stable component detection is a very popular method for feature analysis in images, mainly due to its low computation cost and high repeatability. With the recent advance of feature-based methods in geometric shape analysis, there is significant interest in finding analogous approaches in the 3D world. In this paper, we formulate a diffusion-geometric framework for stable component detection in non-rigid 3D shapes, which can be used for geometric feature detection and description. A quantitative evaluation of our method on the SHREC'10 feature detection benchmark shows its potential as a source of high-quality features.

## 1. Introduction

Over the past decade, feature-based methods have become a ubiquitous tool in image analysis and a *de facto* standard in many computer vision and pattern recognition problems. More recently, there has been an increased interest in developing similar methods for the analysis of 3D shapes. Feature descriptors play an important role in many shape analysis applications, such as finding shape correspondence [31] or assembling fractured models [11] in computational archaeology. Bags of features [28, 24, 32] and similar approaches [21] were introduced as a way to construct global shape descriptors that can be efficiently used for large-scale shape retrieval.

Many shape feature detectors and descriptors draw inspiration from and follow analogous methods in image analysis. For example, detection of geometric structures analogous to corners [27] and edges [14] in images has been studied. The histogram of intrinsic gradients used in [35] is similar in principle to the scale invariant feature transform (SIFT) [16] which has recently become extremely popular in image analysis. In [10], the integral invariant signatures [17] successfully employed in 2D shape analysis were extended to 3D shapes.

Examples of 3D-specific descriptors include the popular spin image [12], based on representation of the shape normal field in a local system of coordinates. Recent studies introduced versatile and computationally efficient descriptors based on the heat kernel [30, 3] describing the local heat propagation properties on a shape. The advantage of these methods is the fact that heat diffusion geometry is intrinsic and thus deformation-invariant, which makes descriptors based on it applicable in deformable shape analysis.

### 1.1. Related work

A different class of feature detection methods tries to find stable components or regions in the analyzed image or shape. In the image processing literature, the watershed transform is the

precursor of many algorithms for stable component detection [6, 33]. In the computer vision and image analysis community, stable component detection is used in the maximally stable extremal regions (MSER) algorithm [18]. MSER represents intensity level sets as a component tree and attempts finding 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. Alternative stability criteria based on geometric scale-space analysis have been recently proposed in [13].

In the shape analysis community, shape decomposition into characteristic primitive elements was explored in [22]. Methods similar to MSER have been explored in the works on topological persistence [8]. Persistence-based clustering [4] was used by Skraba *et al.* [29] to perform shape segmentation. In [7], Digne *et al.* 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.

### 1.2. Main contribution

The main contribution of our framework is three-fold. First, in Section 2 we introduce a generic framework for stable component detection, which unites vertex- and edge-weighted graph representations (as opposed to vertex-weighting used in image and shape maximally stable component detectors [18, 7]). Our results (see Section 4) show that the edge-weighted formulation is more versatile and outperforms its vertex-weighted counterpart in terms of feature repeatability. Second, in Section 3 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. Third, in Section 4 we show a comprehensive evaluation of different settings of our method on a standard feature detection benchmark comprising

<sup>1</sup>School of Electrical Engineering, Tel Aviv University.

<sup>2</sup>Institute of Computational Science, University of Lugano.

shapes undergoing a variety of transformations (also see Figures 1 and 2).

## 2. Diffusion geometry

Diffusion geometry is an umbrella term referring to geometric analysis of diffusion or random walk processes. We model a shape as a compact two-dimensional Riemannian manifold  $X$ . In its simplest setting, a diffusion process on  $X$  is described by the partial differential equation

$$\left(\frac{\partial}{\partial t} + \Delta\right)f(t, x) = 0, \quad (1)$$

called the *heat equation*, where  $\Delta$  denotes the positive-semidefinite Laplace-Beltrami operator associated with the Riemannian metric of  $X$ . The heat equation describes the propagation of heat on the surface and its solution  $f(t, x)$  is the heat distribution at a point  $x$  in time  $t$ . The initial condition of the equation is some initial heat distribution  $f(0, x)$ ; if  $X$  has a boundary, appropriate boundary conditions must be added.

The solution of (1) corresponding to a point initial condition  $f(0, x) = \delta(x, y)$ , is called the *heat kernel* and represents the amount of heat transferred from  $x$  to  $y$  in time  $t$  due to the diffusion process. The value of the heat kernel  $h_t(x, y)$  can also be interpreted as the transition probability density of a random walk of length  $t$  from the point  $x$  to the point  $y$ .

Using spectral decomposition, the heat kernel can be represented as

$$h_t(x, y) = \sum_{i \geq 0} e^{-\lambda_i t} \phi_i(x) \phi_i(y). \quad (2)$$

Here,  $\phi_i$  and  $\lambda_i$  denote, respectively, the eigenfunctions and eigenvalues of the Laplace-Beltrami operator satisfying  $\Delta \phi_i = \lambda_i \phi_i$  (without loss of generality, we assume  $\lambda_i$  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  $X$ , its eigenfunctions and eigenvalues as well as the heat kernel are invariant under isometric transformations (bending) of the shape.

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 (2) is obtained,

$$c(x, y) = \sum_{i \geq 1} \frac{1}{\lambda_i} \phi_i(x) \phi_i(y). \quad (3)$$

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.

By setting  $y = x$ , both the heat and the commute time kernels,  $h_t(x, x)$  and  $c(x, x)$  express the probability density of remaining at a point  $x$ , respectively after time  $t$  and after any time. The value  $h_t(x, x)$ , sometimes referred to as the *auto-diffusivity function*, is related to the Gaussian curvature  $K(x)$  through

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

This relation coincides with the well-known fact that heat tends to diffuse slower at points with positive curvature, and faster at points with negative curvature.

For any  $t > 0$ , the values of  $h_t(x, y)$  at every  $x$  and  $y \in B_\epsilon(x)$  in a small neighborhood around  $x$  contain full information about the intrinsic geometry of the shape. Furthermore, Sun *et al.* [30] show that under mild technical conditions, the set  $\{h_t(x, x)\}_{t>0}$  is also fully informative (note that the auto-diffusivity function has to be evaluated at all values of  $t$  in order to contain full information about the shape metric).

### 2.1. Numerical computation

In the discrete setting, we assume that the shape is sampled at a finite number of points  $V = \{v_1, \dots, v_N\}$ , upon which a simplicial complex (triangular mesh) with vertices  $V$ , edges  $E \subset V \times V$  and faces  $F \subset V \times V \times V$  is constructed. The computation of the discrete heat kernel  $h_t(v_1, v_2)$  and the associated diffusion geometry constructs is performed using formula (2), 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 (FEM) [26], or by discretization of the Laplace operator on the mesh followed by its eigendecomposition. Here, we adopt the second approach according to which the discrete Laplace-Beltrami operator is expressed in the following generic form,

$$(\Delta_X f)_i = \frac{1}{a_i} \sum_j w_{ij} (f_i - f_j), \quad (5)$$

where  $f_i = f(v_i)$  is a scalar function defined on  $V$ ,  $w_{ij}$  are weights, and  $a_i$  are normalization coefficients. In matrix notation, (5) can be written as  $\Delta_X f = A^{-1} W f$ , where  $f$  is an  $N \times 1$  vector,  $A = \text{diag}(a_i)$  and  $W = \text{diag}(\sum_{l \neq i} w_{il}) - (w_{ij})$ . The discrete eigenfunctions and eigenvalues are found by solving the *generalized eigendecomposition* [15]  $W \Phi = A \Phi \Lambda$ , where  $\Lambda = \text{diag}(\lambda_l)$  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 [9, 34]. For triangular meshes, a popular choice adopted in this paper is the *cotangent weight* scheme [25, 19], in which

$$w_{ij} = \begin{cases} (\cot \alpha_{ij} + \cot \beta_{ij})/2 & (v_i, v_j) \in E; \\ 0 & \text{else,} \end{cases} \quad (6)$$

where  $\alpha_{ij}$  and  $\beta_{ij}$  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.

## 3. Maximally stable components

Let us now focus on the undirected graph with the vertex set  $V$  and edge set  $E$  underlying the discretization of a shape, which with some abuse of notation we will henceforth denote as  $X = (V, E)$ . We say that two vertices  $v_1$  and  $v_2$  are *adjacent* if  $(v_1, v_2) \in E$ . An ordered sequence  $\pi = \{v_1, \dots, v_k\}$  of vertices is

called a *path* if for any  $i = 1, \dots, k-1$ ,  $v_i$  is adjacent to  $v_{i+1}$ . In this case, we say that  $v_1$  and  $v_k$  are linked in  $X$ . The graph is said to be *connected* if every pair of vertices in it is linked. A graph  $Y = (V' \subseteq V, E' \subseteq E)$  is called a *subgraph* of  $X$  and denoted by  $Y \subseteq X$ . We say that  $Y$  is a (connected) *component* of  $X$  if  $Y$  is a connected subgraph of  $X$  that is maximal for this property (i.e., for any connected subgraph  $Z$ ,  $Y \subseteq Z \subseteq X$  implies  $Y = Z$ ). Given  $E' \subseteq E$ , the graph induced by  $E'$  is the graph  $Y = (V', E')$  whose vertex set is made of all vertices belonging to an edge in  $E'$ , i.e.,  $V' = \{v \in V : \exists v' \in V, (v, v') \in E'\}$ .

A scalar function  $f : V \rightarrow \mathbb{R}$  is called a *vertex weight*, and a graph equipped with it is called *vertex-weighted*. Similarly, a graph equipped with a function  $d : E \rightarrow \mathbb{R}$  defined on the edge set is called *edge-weighted*. In what follows, we will assume both types of weights to be non-negative. Grayscale images are often represented as vertex-weighted graphs with some regular (e.g., four-neighbor) connectivity and weights corresponding to the intensity of the pixels. Edge weights can be obtained, for example, by considering a local distance function measuring the dissimilarity of pairs of adjacent pixels. While vertex weighting is limited to scalar (grayscale) images, edge weighting is more general.

### 3.1. Component trees

Let  $(X, f)$  be a vertex-weighted graph. For  $\ell \geq 0$ , the  $\ell$ -*cross-section* of  $X$  is defined as the graph induced by  $E_\ell = \{(v_1, v_2) \in E : f(v_1), f(v_2) \leq \ell\}$ . Similarly, a cross-section of an edge-weighted graph  $(X, d)$  is induced by the edge subset  $E_\ell = \{e \in E : d(e) \leq \ell\}$ . A connected component of the cross-section is called an  $\ell$ -*level set* of the weighted graph.

For any component  $C$  of  $X$ , we define the *altitude*  $\ell(C)$  as the minimal  $\ell$  for which  $C$  is a component of the  $\ell$ -cross-section of  $X$ . 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*. Note that the above definitions are valid for both vertex- and edge-weighted graphs.

### 3.2. Maximally stable components

Since in our discussion undirected graphs are used as a discretization of smooth manifolds, we can associate with every component  $C$  (or every subset of the vertex set in general) a measure of *area*,  $A(C)$ . In the simplest setting, the area of  $C$  can be thought of as its cardinality. In a better discretization, each vertex  $v$  in the graph is associated with a discrete area element  $da(v)$ , and the area of a component is defined as

$$A(C) = \sum_{v \in C} da(v). \quad (7)$$

Let now  $\{(\ell, C_\ell)\}$  be a sequence of nested components forming a branch in the component tree. We define the *instability* of  $C_\ell$  as

$$s(\ell) = \frac{dA(C_\ell)}{d\ell}. \quad (8)$$

In other words, the more the area of a component changes with the change of  $\ell$ , the less stable it is. A component  $C_{\ell^*}$  is called *maximally stable* if the instability function has a local minimum at  $\ell^*$ . Maximally stable components are widely known in the computer vision literature under the name of *maximally stable extremal regions* or *MSERs* for short [18], with  $s(\ell^*)$  usually referred to as the *region score*.

It is important to note that in their original definition, MSERs were defined on a component tree of a vertex-weighted graph, while our definition is more general and allows for edge-weighted graphs as well. The importance of such an extension will become evident in the sequel. Also, the original MSER algorithm [18] assumes the vertex weights to be quantized, while our formulation is suitable for scalar fields whose dynamic range is unknown *a priori*.

### 3.3. Computational aspects

We use the quasi-linear time algorithm detailed in [23] for the construction of vertex-weighted component trees, and its straightforward adaptation to the edge-weighted case. 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  $O(N \log \log N)$ .

The derivative (8) of the component area with respect to  $\ell$  constituting the stability function is computed using finite differences in each branch of the tree. For example, in a branch  $C_{\ell_1} \subseteq C_{\ell_2} \subseteq \dots \subseteq C_{\ell_K}$ ,

$$s(\ell_k) \approx \frac{A(C_{\ell_{k+1}}) - A(C_{\ell_{k-1}})}{\ell_{k+1} - \ell_{k-1}}. \quad (9)$$

The function is evaluated and its local minima are detected in a single pass over the branches of the component tree starting from the leaf nodes. We further filter out maximally stable regions with too high values of  $s$ . In cases where two nested regions overlapping by more than a predefined threshold are detected as maximally stable, only the bigger one is kept.

## 4. Weighting functions

Unlike images where methods based on the analysis of the component tree have been shown to be extremely successful e.g. for segmentation or affine-invariant feature detection (namely, the MSER feature detector), similar techniques have been only scarcely explored for 3D shapes (with the notable exceptions of [7] and [29]). One of possible reasons is the fact that while images readily offer pixel intensities as the trivial vertex weight field, 3D shapes are not generally equipped with any such field. While the use of the mean curvature was proposed in [7], it lacks most of invariance properties required in deformable shape analysis. Here, we follow [29] in adopting the diffusion geometry framework and show that it allows to construct both vertex and edge weights suitable for the definition of maximally stable components with many useful properties.

Given a vertex  $v$ , the values of the discrete auto-diffusivity function can be directly used as the vertex weights,

$$f(v) = h_t(v, v). \quad (10)$$

Maximally stable components defined this way are intrinsic and, thus, invariant to non-rigid bending. Such strong invariance properties are particularly useful in the analysis of deformable shapes. However, unlike images where the intensity field contains all information about the image, the above weighting function does not describe the intrinsic geometry of the shape entirely. It furthermore depends on the selection of the scale parameter  $t$ .

Edge weights constitute a more flexible alternative allowing to incorporate fuller geometric information. The simplest edge weighting scheme can be obtained from a vector-valued field defined on the vertices of the graph. For example, associating  $h_t(v, v)$  for  $t \in [t_1, t_2]$  with each vertex  $v$ , one can define an edge weighting function

$$\begin{aligned} d(v_1, v_2) &= \|h_t(v_1, v_1) - h_t(v_2, v_2)\|_t \\ &= \left( \int_{t_1}^{t_2} (h_t(v_1, v_1) - h_t(v_2, v_2))^2 dt \right)^{1/2} \end{aligned} \quad (11)$$

(here, we write  $\|\cdot\|_t$  to make explicit that the norm is taken with respect to the variable  $t$ ). The function has a closed-form expression that can be obtained by substituting the spectral decomposition (2) of the heat kernel. The advantage of this approach stems from its ability to incorporate multiple scales. Theoretically, the set of  $h_t(v, v)$  at all scales contains full information about the intrinsic geometry of the shape.

In a more general setting, edge weights do not necessarily need to stem from any finite- or infinite-dimensional vector field defined on the vertices. For example, since the discrete heat kernel  $h_t(v_1, v_2)$  represents “proximity” between  $v_1$  and  $v_2$ , a function inversely proportional to the value of the heat kernel, e.g.,

$$d(v_1, v_2) = \frac{1}{h_t(v_1, v_2)} \quad (12)$$

can be used as an edge weight. For sufficiently small values of  $t$ , this function also contains full information about the shape’s intrinsic geometry.

Another way of creating edge weights inversely proportional to  $h_t$  is by integrating the squared difference between the kernels centered at  $v_1$  and  $v_2$  over the entire shape,

$$\begin{aligned} d(v_1, v_2) &= \|h_t(v_1, \cdot) - h_t(v_2, \cdot)\|_X \\ &= \left( \sum_{v \in V} (h_t(v_1, v) - h_t(v_2, v))^2 da(v) \right)^{1/2}. \end{aligned} \quad (13)$$

This construction has been previously introduced in [5] under the name of *diffusion distance*, which constitutes an intrinsic metric on  $X$  and is fully informative for small  $t$ ’s.

#### 4.1. Scale invariance

The vertex weighting function (10) and the edge weighting functions (11), (12) and (13) based on the heat kernel are not

scale invariant since a global scaling of the shape by a factor  $\gamma > 0$  influences the heat kernel as  $\gamma^2 h_{\gamma^2 t}(v_1, v_2)$ , scaling by  $\gamma^2$  both the time parameter and the kernel itself. A possible remedy is to replace the heat kernel by the scale invariant commute time kernel. However, due to the slow decay of the expansion coefficients  $\lambda_i^{-1}$  in (3) compared to  $e^{-\lambda_i t}$  in (2), the numerical computation of the commute time kernel is more difficult as it requires many more eigenfunctions of the Laplacian to achieve the same accuracy.

As an alternative, it is possible to use a sequence of transformations of  $h_t(v_1, v_2)$  that renders it scale invariant [3]. First, the heat kernel is sampled logarithmically in time. Next, the logarithm and a derivative with respect to time of the heat kernel values are taken to undo the multiplicative constant. Finally, taking the magnitude of the Fourier transform allows to undo the scaling of the time variable. This yields the *modified heat kernel* of the form

$$\hat{h}_\omega(v_1, v_2) = \left| \mathcal{F} \left\{ \frac{\partial \log h_t(v_1, v_2)}{\partial \log t} \right\}(\omega) \right|, \quad (14)$$

where  $\omega$  denotes the frequency variable of the Fourier transform. The transform is computed numerically using the FFT as detailed in [3]. Substituting  $\hat{h}_\omega$  into (11)–(12) yields scale invariant edge weighting functions.<sup>3</sup> By selecting a single frequency  $\omega$ , one can construct a scale invariant vertex weight  $f(v) = \hat{h}_\omega(v, v)$  similar to (10). Another way of constructing a scale invariant vertex weight is by integrating  $\hat{h}_\omega$  over a range of frequencies, e.g.,

$$f(v) = \|\hat{h}_\omega(v, v)\|_\omega = \left( \int_{\omega_1}^{\omega_2} \hat{h}_\omega^2(v, v) d\omega \right)^{1/2}. \quad (15)$$

## 5. Descriptors

### 5.1. Point descriptors

Once the regions are detected, their content can be described using any standard point-wise descriptor of the form  $\alpha : V \rightarrow \mathbb{R}^q$ . In particular, here we consider point-wise *heat kernel descriptors* proposed in [30]. The heat kernel descriptor (or *heat kernel signature*, HKS) is computed by taking the values of the discrete auto-diffusivity function at vertex  $v$  at multiple times,  $\alpha(v) = (h_{t_1}(v, v), \dots, h_{t_q}(v, v))$ , where  $t_1, \dots, t_q$  are some fixed time values. Such a descriptor is a vector of dimensionality  $q$  at each point. Since the heat kernel is an intrinsic quantity, the HKS is invariant to isometric transformations of the shape.

A scale-invariant version of the HKS descriptor (SI-HKS) can be obtained as proposed [3] by replacing  $h_t$  with  $\hat{h}_\omega$

<sup>3</sup>Since the component inclusion relations giving rise to component tree are invariant to any monotonous transformation of the weighting functions, it is sufficient to undo just undoing the scaling of the time parameter  $t$  without undoing the scaling of the kernel itself. However, such a transformation affects the scores of the detected regions. We found that the logarithmic transformation and derivative improve repeatability. Furthermore, by completely undoing the effect of scaling, the modified heat kernel can be used both in the weighting function and in descriptors of the maximally stable components as detailed in the following section.



Figure 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 Figure 2). Regions were detected using  $h_t(v, v)$  as vertex weight function, with  $t = 2048$ .

from (14), yielding  $\alpha(v) = (\hat{h}_{\omega_1}(v, v), \dots, \hat{h}_{\omega_q}(v, v))$ , where  $\omega_1, \dots, \omega_q$  are some fixed frequency values. In the following experiments, the heat kernel was sampled at time values  $t = 2^1, 2^{1+1/16}, \dots, 2^{25}$ . The first six discrete frequencies of the Fourier transform were taken, repeating the settings of [3].

### 5.2. Region descriptors

Given a descriptor  $\alpha(v)$  at each vertex  $v \in V$ , the simplest way to define a *region descriptor* of a component  $C \subset V$  is by computing the average of  $\alpha$  in  $C$ ,

$$\beta(C) = \sum_{v \in C} \alpha(v) da(v). \quad (16)$$

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

An alternative construction considered here follows Ovsjanikov *et al.* [24] where a global shape descriptors were obtained from point-wise descriptors using the *bag of features* paradigm [28]. In this approach, a fixed “geometric vocabulary”  $\alpha_1, \dots, \alpha_p$  is computed by means of an off-line clustering of the descriptor space. Next, each point descriptor at  $v$  is represented in the vocabulary using vector quantization, yielding a point-wise  $p$ -dimensional distribution of the form

$$\theta(v) \propto e^{-\|\alpha(v) - \alpha_i\|^2 / 2\sigma^2}. \quad (17)$$

The distribution is normalized in such a way that the elements of  $\theta(v)$  sum to one. In the case of  $\sigma = 0$ , hard vector quantization is used, and  $\theta_i(v) = 1$  for  $\alpha_i$  being the closest element of the geometric vocabulary to  $\alpha(v)$  in the descriptor space, and zero elsewhere. Given a component  $C$ , we can define a *local bag of features* by computing the distribution of geometric words over the region,

$$\beta(C) = \sum_{v \in C} \theta(v) da(v). \quad (18)$$

Such a bag of features is used as a region descriptor of dimensionality  $p$ .

## 6. Results

### 6.1. Dataset

The proposed approach was tested on the data of the SHREC’10 feature detection and description benchmark [2]. The SHREC dataset consisted of three shape classes, with simulated transformations applied to them. Shapes are represented as triangular meshes with approximately 10,000 to 50,000 vertices. In our experiments, all meshes were downsampled to at most 10,000 vertices. Each shape class contained 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). In transformation appeared in five different strengths. 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 groundtruth correspondence and its symmetric counterpart were used.

We also used several deformable shapes from the TOSCA dataset [1] for a qualitative evaluation.

### 6.2. Detector repeatability

The evaluation of the proposed feature detector and descriptor followed the spirit of the influential work by Mikolajczyk *et al.* [20]. In the first experiment, the repeatability of the detector was evaluated. Let  $X$  and  $Y$  be the null and the transformed version of the same shape, respectively. Let  $X_1, \dots, X_m$  and  $Y_1, \dots, Y_n$  denote the regions detected in  $X$  and  $Y$ , and let  $X'_j$  be the image of the region  $Y_j$  in  $X$  under the ground-truth correspondence.<sup>4</sup> Given two regions  $X_i$  and  $Y_j$ , their *overlap* is defined as the following area ratio

$$O(X_i, X'_j) = \frac{A(X_i \cap X'_j)}{A(X_i \cup X'_j)} = \frac{A(X_i \cap X'_j)}{A(X_i) + A(X'_j) - A(X_i \cap X'_j)}. \quad (19)$$

<sup>4</sup>As some of the transformed shapes had missing data compared to the null shape, comparison was defined single-sidedly. Only regions in the transformed shape that had no corresponding regions in the null counterpart decreased the overlap score, while unmatched regions of the null shape did not.



Figure 2: Maximally stable regions detected on shapes from the SHREC'10 dataset using the vertex weight  $h_t(v, v)$  with  $t = 2048$ . First row: different approximate isometries of the human shape. Second row: different transformations (left-to-right): holes, localscale, noise, shotnoise and scale.

The *repeatability at overlap  $o$*  is defined as the percentage of regions in  $Y$  that have corresponding counterparts in  $X$  with overlap greater than  $o$  [20]. An ideal detector has the repeatability of 1.

Four vertex weight functions were compared: discrete heat kernel (10) with  $t = 2048$ , commute time kernel (3), modified heat kernel with  $\omega = 0$ , and the norm of the modified heat kernel (15). These four scalar fields were also used to construct edge weights according to  $d(v_1, v_2) = |f(v_1) - f(v_2)|$ . Furthermore, since these kernels are functions of a pair of vertices, they were used to define edge weights according to (12). In addition, we also tested edge weights constructed according to (11) and the diffusion distance (13). Unless mentioned otherwise,  $t = 2048$  was used for the heat kernel and  $\omega = 0$  for the modified heat kernel, as these settings turned out to give best performance on the SHREC'10 dataset.

We first evaluated different region detectors qualitatively using shapes from the SHREC'10 and the TOSCA datasets. Figure 1 shows the regions detected using the vertex weight  $h_t(v, v)$  with  $t = 2048$  on a few sample shapes from the TOSCA dataset. Figures 2 and 5 depict the maximally stable components detected with the same settings on several shapes from the SHREC dataset. Figure 3 shows the regions obtained using the edge weighting function  $1/h_t(v_1, v_2)$ . In all cases, the detected regions appear robust and repeatable under the transformations. Surprisingly, many of these regions have a clear semantic interpretation. Moreover, similarly looking regions are detected on the male and female shapes, and the upper half of the centaur. This makes the proposed feature detector a good candidate for



Figure 5: The set of maximally stable regions extracted from one of the shapes in Figure 2.

partial shape matching and retrieval.

In order to select optimal cutoff threshold of the instability function (i.e., the maximum region instability value that is still accepted by the detector), we estimated the empirical distributions of the detecting regions as a function of the instability score and their overlap with the corresponding groundtruth regions. These histograms are depicted in Figure 4. A good detector should produce many regions with overlap close to 100% that have low instability, and produce as few as possible low-overlap regions that have very high instability that can be separated from the high-overlap regions by means of a threshold. In



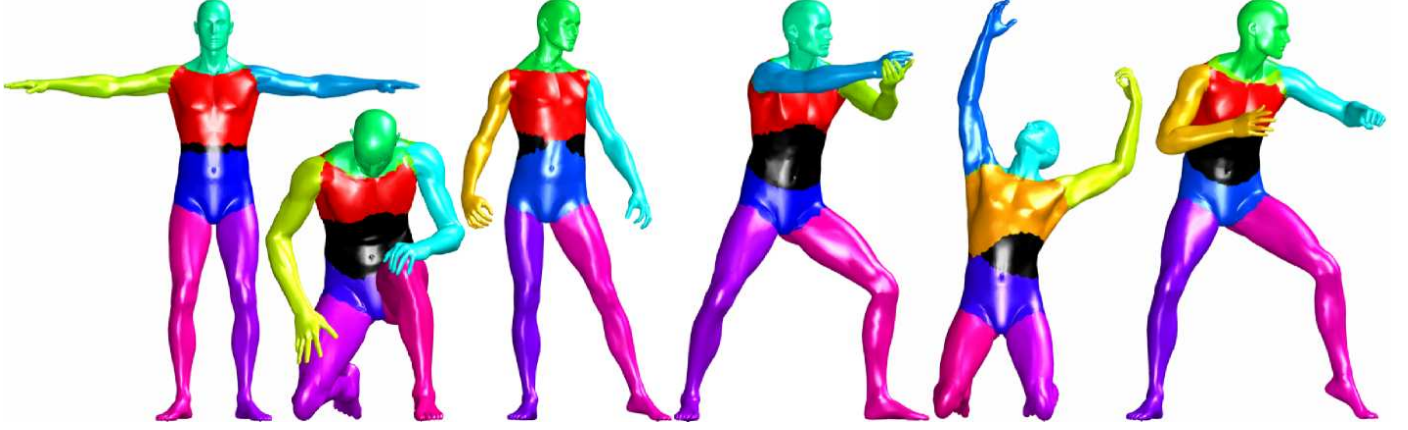


Figure 3: Maximally stable regions detected on shapes from the SHREC'10 dataset using the edge weight  $1/h_t(v_1, v_2)$  with  $t = 2048$ . Region coloring is arbitrary.

each of the tested detectors, the instability score threshold was selected to maximize the detection of high-overlap regions.

Table 1 summarizes the repeatability of different weighting functions at overlap of 75%. Figures 6 and 7 depict the repeatability and the number of correctly matching regions as the function of the overlap for the best four of the compared weighting functions. We conclude that scale-dependent weighting generally outperform their scale-invariant counterparts in terms of repeatability. The four scalar fields corresponding to different 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 is also the edge weight  $1/c(v_1, v_2)$ .

### 6.3. Descriptor discriminativity

In the second experiment, the discriminativity of region descriptors was evaluated by measuring the relation between distance in the descriptor space and the overlap between the corresponding regions.

Using the notation from the previous section, let  $Y_i$  be one of the  $n$  maximally stable components detected on a transformed shape  $Y$ ,  $X'_i$  its image on the null shape  $X$  under the ground truth correspondence, and let  $X_j$  denote one of the  $m$  maximally stable components detected on the null shape. A groundtruth relation between the regions is established by fixing a minimum overlap  $\rho = 0.75$  and deeming  $Y_i$  and  $X_j$  matching if  $o_{ij} = O(X'_i, X_j) \geq \rho$ . Let us now be given a region descriptor  $\beta$ ; for simplicity we assume the distance between the descriptors to be the standard Euclidean distance. By setting a threshold  $\tau$  on this distance,  $Y_i$  and  $X_j$  will be classified as positives if  $d_{ij} = \|\beta(Y_i) - \beta(X_j)\| \leq \tau$ . We define the *true positive rate* as the ratio

$$\text{TPR} = \frac{|\{d_{ij} \leq \tau\}|}{|\{o_{ij} \geq \rho\}|}, \quad (20)$$

similarly, the *false positive rate* is defined as

$$\text{FPR} = \frac{|\{d_{ij} > \tau\}|}{|\{o_{ij} < \rho\}|}. \quad (21)$$

A related quantity is the *false negative rate* defined as  $\text{FNR} = 1 - \text{TPR}$ . By varying the threshold  $\tau$ , a set of pairs (FPR, TPR) referred to as the *receiver operator characteristic* (ROC) curve is obtained. The particular point on the ROC curve for which the false positive and false negative rates coincide is called *equal error rate* (EER). We use EER as a scalar measure for the descriptor discriminativity. Ideal descriptors have  $\text{EER} = 0$ .

Another descriptor performance criterion used here considers the first matches produced by the descriptor distance. For that purpose, for each  $X_i$  we define its first match as the  $Y_{j^*(i)}$  with  $j^*(i) = \arg \min_j d_{ij}$  (nearest neighbor of  $\beta(X_i)$  in the descriptor space). The *matching score* is defined as the ratio of correct first matches for a given overlap  $\rho$ ,

$$\text{score}(\rho) = \frac{|\{o_{ij^*(i)} \geq \rho\}|}{m}. \quad (22)$$

The following four weighting functions exhibiting best repeatability scores in the previous experiment were used to define region detectors: the edge weight  $1/h_t(v_1, v_2)$  with  $t = 1024$  (absolute winner in terms of repeatability), the vertex weight  $h_t(v, v)$  (second-best repeatability), its edge-weight counterpart  $|h_t(v_1, v_1) - h_t(v_2, v_2)|$  (gives lower repeatability scores but supplies almost twice correspondences), and the edge weight  $1/c(v_1, v_2)$  (best scale invariant detector). Given the maximally stable components detected by a selected detector, region descriptors were calculated. We used two types of point descriptors: the heat kernel signature  $h_t(v, v)$  sampled at six time values  $t = 16, 22.6, 32, 45.2, 64, 90.5, 128$ , and its scale invariant version  $\hat{h}_\omega(v, v)$ , for which we have taken the first six discrete frequencies of the Fourier transform (these are settings identical to [3]). These point descriptors were used to create region descriptors using averaging () and local bags of features (). Bags of features were tested with vocabulary sizes  $p = 10$  and  $12$ . Table 2 summarizes the performance in terms of EER of different combinations of weighting functions and region descriptors. Figure 8 depicts the ROC curves of different descriptors of vertex- and edge-weighted maximally stable component detectors.

Figures 9–12 show the number of correct first matches and the matching score as a function of the overlap for different

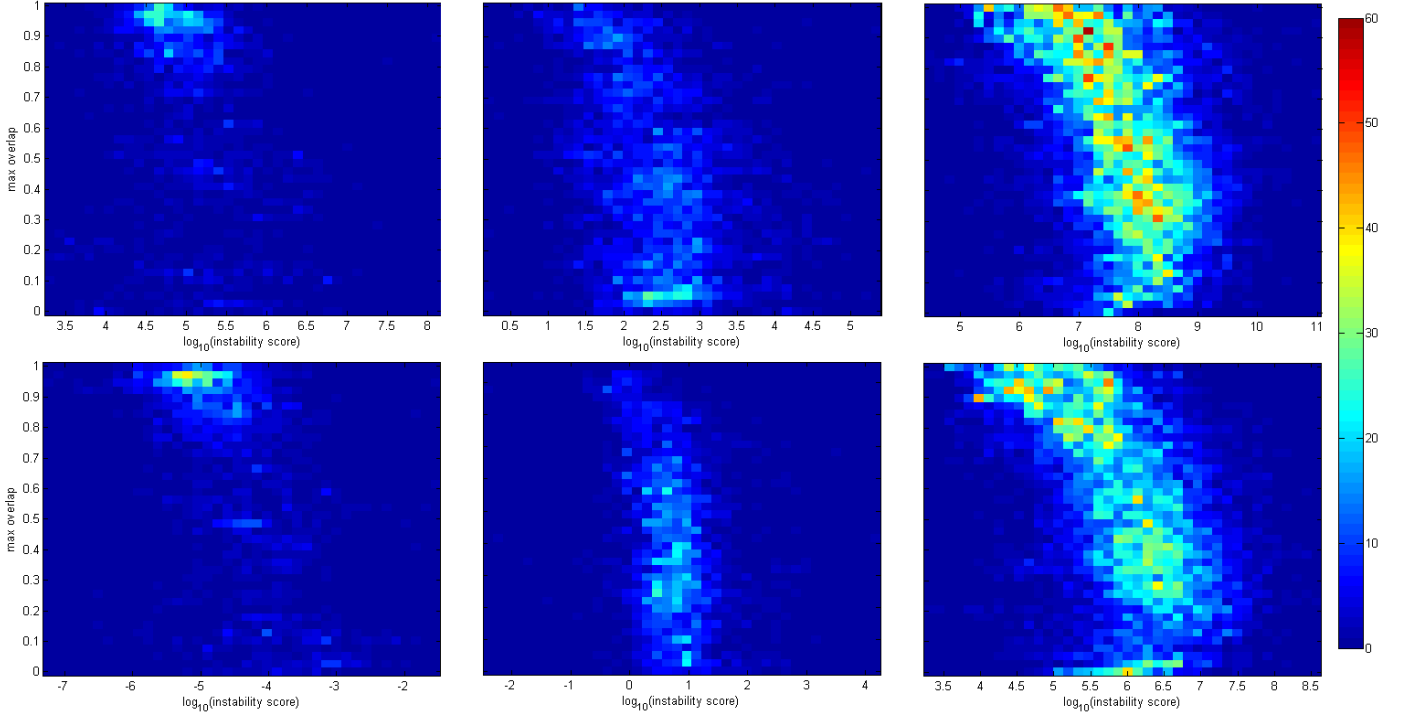


Figure 4: Distributions of maximally stable components as a function of the overlap to the groundtruth regions and instability score. Left-to-right top-to-bottom are shown the following weighting function: vertex weight  $h_t(v, v)$  at  $t = 2048$ , vertex weight  $c(v, v)$ , edge weight  $\|h_t(v_1, \cdot) - h_t(v_2, \cdot)\|_X$  (diffusion-distance) at  $t = 2048$ , edge weight  $1/h_t(v_1, v_2)$  at  $t = 2048$ , vertex weight  $\hat{h}_\omega(v, v)$  at  $\omega = 0$  and edge weight  $|h_t(v_1, v_1) - h_t(v_2, v_2)|$  at  $t = 2048$ . Good detectors are characterized by a large number of high-overlap stable regions (many regions in the upper left corner of the plot) that can be easily separated from the low-overlap regions that should be concentrated in the lower right corner.

choices of weighting functions and descriptors. Examples of matching regions are depicted in Figure 13.

We conclude that the scale invariant HKS descriptor consistently exhibits higher performance in both the average and bag of features flavors. The latter flavors perform approximately the same. The HKS descriptor, on the other hand, performs better in the bag of feature setting, though never reaching the scores of SIHKS. Surprisingly, as can be seen from Figures 9–12, the SIHKS descriptor is consistently more discriminative even in transformations not including scaling.

## 7. Conclusions

We presented a generic framework for the detection of stable regions in non-rigid shapes. Our approach is based on the maximization of a stability criterion in a component tree representation of the shape with vertex or edge weights. Using diffusion geometric weighting functions allows obtaining a feature detection algorithm that is invariant to a wide class of shape transformations, in particular, non-rigid bending and global scaling, which makes our approach applicable in the challenging setting of deformable shape analysis. In followup studies, we are going to explore the uses of the proposed feature detectors and descriptors in shape matching and retrieval problems.

## References

- [1] Bronstein, A., Bronstein, M., Bronstein, M., Kimmel, R., 2008. Numerical geometry of non-rigid shapes. Springer.
- [2] Bronstein, A., Bronstein, M., Bustos, B., Castellani, U., Crisani, M., Falcidieno, B., Guibas, L., Kokkinos, I., Murino, V., Ovsjanikov, M., et al., 2010. SHREC 2010: robust feature detection and description benchmark.
- [3] Bronstein, M. M., Kokkinos, I., 2010. Scale-invariant heat kernel signatures for non-rigid shape recognition. In: Proc. CVPR.
- [4] Chazal, F., Guibas, L., Oudot, S., Skraba, P., 2009. Persistence-based clustering in Riemannian manifolds.
- [5] Coifman, R., Lafon, S., 2006. Diffusion maps. Applied and Computational Harmonic Analysis 21 (1), 5–30.
- [6] Couprie, M., Bertrand, G., 1997. Topological grayscale watershed transformation. In: SPIE Vision Geometry V Proceedings. Vol. 3168. pp. 136–146.
- [7] Digne, J., Morel, J., Audfray, N., Mehdi-Souzani, C., 2010. The Level Set Tree on Meshes.
- [8] Edelsbrunner, H., Letscher, D., Zomorodian, A., 2002. Topological persistence and simplification. Discrete and Computational Geometry 28 (4), 511–533.
- [9] Floater, M. S., Hormann, K., 2005. Surface parameterization: a tutorial and survey. Advances in Multiresolution for Geometric Modelling 1.
- [10] Gelfand, N., Mitra, N. J., Guibas, L. J., Pottmann, H., 2005. Robust global registration. In: Proc. SGP.
- [11] Huang, Q., Flöry, S., Gelfand, N., Hofer, M., Pottmann, H., 2006. Re-assembling fractured objects by geometric matching. ACM Trans. Graphics 25 (3), 569–578.
- [12] Johnson, A. E., Hebert, M., 1999. Using spin images for efficient object recognition in cluttered 3D scenes. Trans. PAMI 21 (5), 433–449.
- [13] Kimmel, R., Zhang, C., Bronstein, A. M., Bronstein, M. M., 2010. Are MSER features really interesting? IEEE Trans. PAMI.
- [14] Kolomenkin, M., Shimshoni, I., Tal, A., 2009. On edge detection on surfaces. In: Proc. CVPR.



Weighting function	Maximal instability score used	Avg. number of detected regions	Num. of correspondences at overlap= 75%	Repeatability at overlap= 75%
$h_t(v, v), t = 2048$	$\infty$	11.5	6.89	65%
$c(v, v)$	$\infty$	28.3	5.01	18%
$\hat{h}_\omega(v, v), \omega = 0$	$\infty$	15.1	5.43	40%
$\ \hat{h}_\omega(v, v)\ _\omega$	$\infty$	20.2	1.96	10%
$ h_t(v_1, v_1) - h_t(v_2, v_2) , t = 2048$	$2.51 \times 10^5$	21.7	12.73	60%
$ c(v_1, v_1) - c(v_2, v_2) $	$\infty$	11.4	4.12	35%
$ \hat{h}_\omega(v_1, v_1) - \hat{h}_\omega(v_2, v_2) , \omega = 0$	$\infty$	71.8	3.33	5%
$1/h_t(v_1, v_2), t = 2048$	$\infty$	13.8	8.64	68%
$1/c(v_1, v_2)$	1	11.1	5.07	45%
$1/\ \hat{h}_\omega(v_1, v_2)\ _\omega$	100	4.8	1.81	45%
$\ h_t(v_1, \cdot) - h_t(v_2, \cdot)\ _X, t = 2048$	$5 \times 10^6$	13.8	7.19	50%
$\ h_t(v_1, v_1) - h_t(v_2, v_2)\ _t, t \in [128, 32 \times 10^3]$	$1.58 \times 10^7$	18.4	9.99	58%

Table 1: Repeatability of maximally stable components with different vertex and edge weighting functions.

Weighting function	HKS Average	HKS BoF( $p = 10$ )	HKS BoF( $p = 12$ )	SI-HKS Average	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

Table 2: Equal error rate (EER) performance of different maximally stable component detectors and descriptors ( $t = 2048$  was used in all cases).  $p$  denotes the vocabulary size in the bag of features region descriptors.

- [15] Lévy, B., 2006. Laplace-Beltrami eigenfunctions towards an algorithm that “understands” geometry. In: Proc. Shape Modeling and Applications.
- [16] Lowe, D., 2004. Distinctive image features from scale-invariant keypoint. IJCV 60 (2), 91–110.
- [17] Manay, S., Hong, B., Yezzi, A., Soatto, S., 2004. Integral invariant signatures. Lecture Notes in Computer Science, 87–99.
- [18] Matas, J., Chum, O., Urban, M., Pajdla, T., 2004. Robust wide-baseline stereo from maximally stable extremal regions. Image and Vision Computing 22 (10), 761–767.
- [19] Meyer, M., Desbrun, M., Schroder, P., Barr, A. H., 2003. Discrete differential-geometry operators for triangulated 2-manifolds. Visualization and Mathematics III, 35–57.
- [20] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L., 2005. A comparison of affine region detectors. IJCV 65 (1), 43–72.
- [21] Mitra, N. J., Guibas, L. J., Giesen, J., Pauly, M., 2006. Probabilistic fingerprints for shapes. In: Proc. SGP.
- [22] Mortara, M., Patane, G., Spagnuolo, M., Falcidieno, B., Rossignac, J., 2003. Blowing bubbles for multi-scale analysis and decomposition of triangle meshes. Algorithmica 38 (1), 227–248.
- [23] Najman, L., Couprie, M., 2006. Building the component tree in quasi-linear time. IEEE Trans. Image Proc. 15 (11), 3531–3539.
- [24] Ovsjanikov, M., Bronstein, A., Guibas, L., Bronstein, M., 2009. Shape Google: a computer vision approach to invariant shape retrieval. In: Proc. NORDIA.
- [25] Pinkall, U., Polthier, K., 1993. Computing discrete minimal surfaces and their conjugates. Experimental mathematics 2 (1), 15–36.
- [26] Reuter, M., Wolter, F.-E., Peinecke, N., 2005. Laplace-spectra as fingerprints for shape matching. In: Proc. ACM Symp. Solid and Physical Modeling. pp. 101–106.
- [27] Sipiran, I., Bustos, B., 2010. A robust 3D interest points detector based on Harris operator. In: Proc. 3DOR. Eurographics, pp. 7–14.
- [28] Sivic, J., Zisserman, A., 2003. Video Google: a text retrieval approach to object matching in videos. In: Proc. CVPR.
- [29] Skraba, P., Ovsjanikov, M., Chazal, F., Guibas, L., 2010. Persistence-based segmentation of deformable shapes. In: Proc. NORDIA. pp. 45–52.
- [30] Sun, J., Ovsjanikov, M., Guibas, L., 2009. A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion. In: Computer Graphics Forum. Vol. 28, pp. 1383–1392.
- [31] Thorstensen, N., Keriven, R., 2009. Non-rigid shape matching using geometry and photometry. In: Proc. CVPR.
- [32] Toldo, R., Castellani, U., Fusiello, A., 2009. Visual vocabulary signature for 3D object retrieval and partial matching. In: Proc. 3DOR.
- [33] Vincent, L., Soille, P., 2002. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. IEEE Trans. PAMI 13 (6), 583–598.
- [34] Wardetzky, M., Mathur, S., Kälberer, F., Grinspun, E., 2008. Discrete Laplace operators: no free lunch. In: Conf. Computer Graphics and Interactive Techniques.
- [35] Zaharescu, A., Boyer, E., Varanasi, K., Horaud, R., 2009. Surface feature detection and description with applications to mesh matching. In: Proc. CVPR.

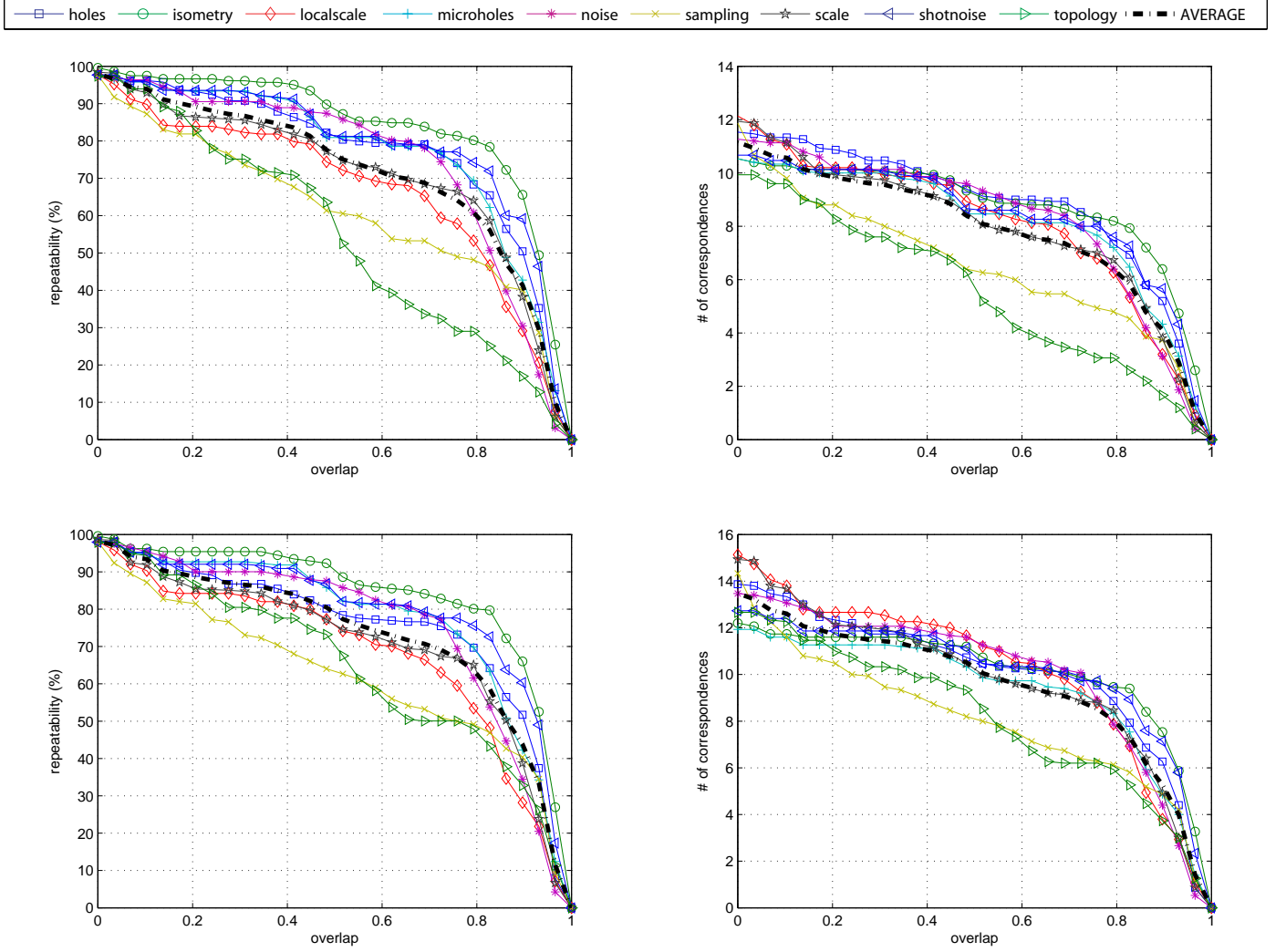


Figure 6: Repeatability of maximally stable components with the vertex weight  $h_t(v, v)$  (first row) and edge weight  $1/h_t(v_1, v_2)$  (second row),  $t = 2048$ .

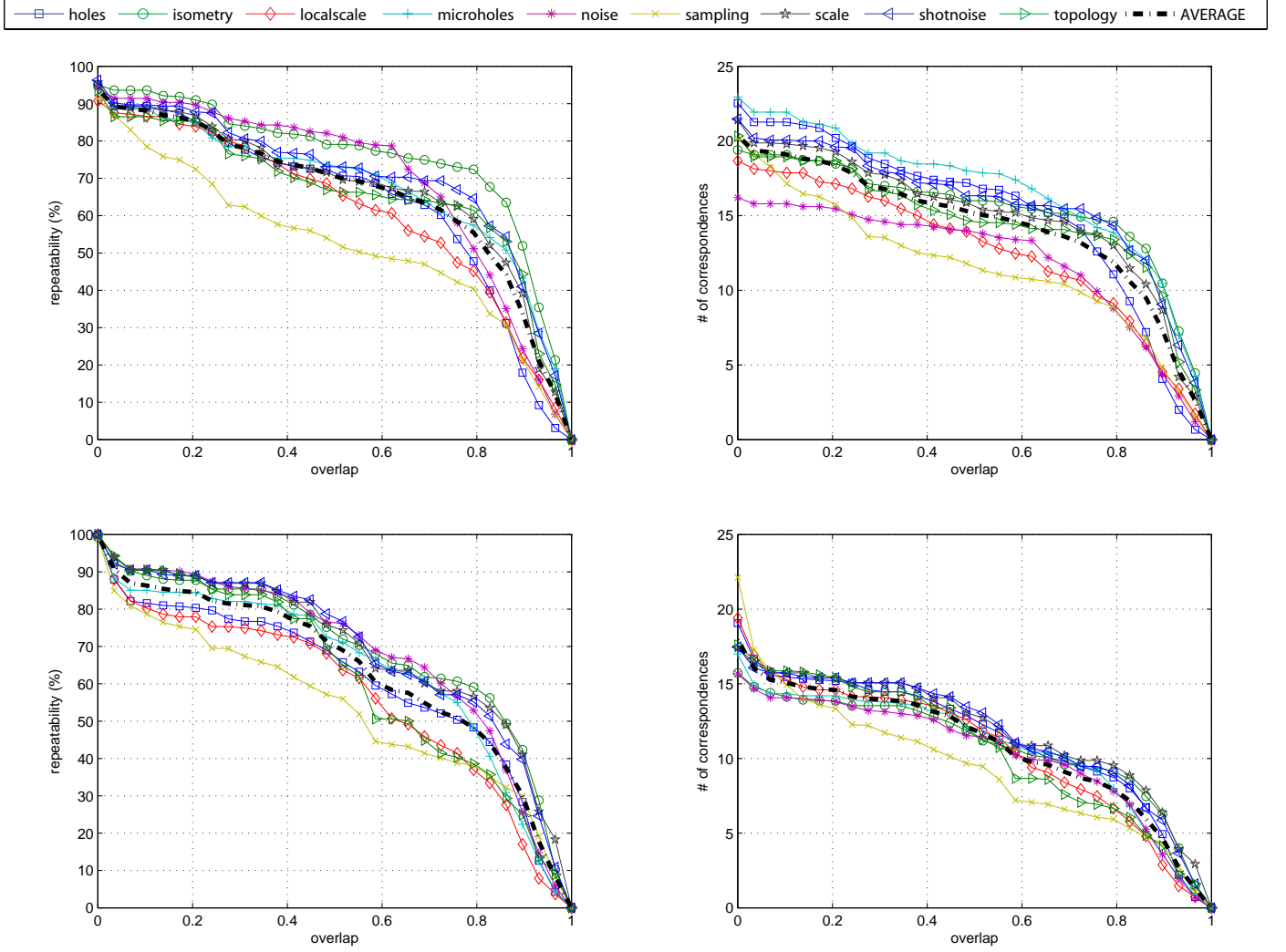


Figure 7: Repeatability of maximally stable components with the edge weight  $|h_t(v_1, v_1) - h_t(v_2, v_2)|$  (first row) and edge weight  $1/c(v_1, v_2)$  (second row),  $t = 2048$ .

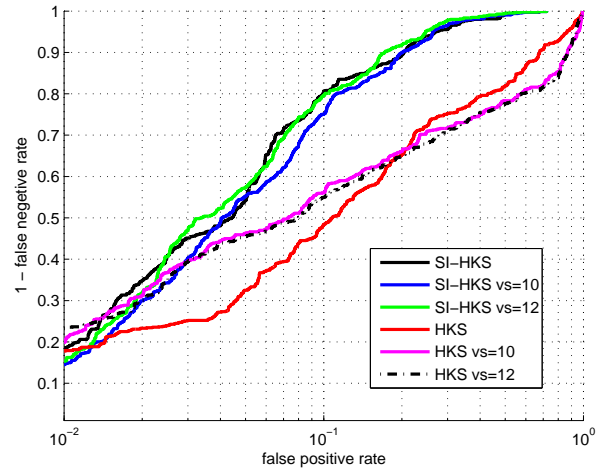
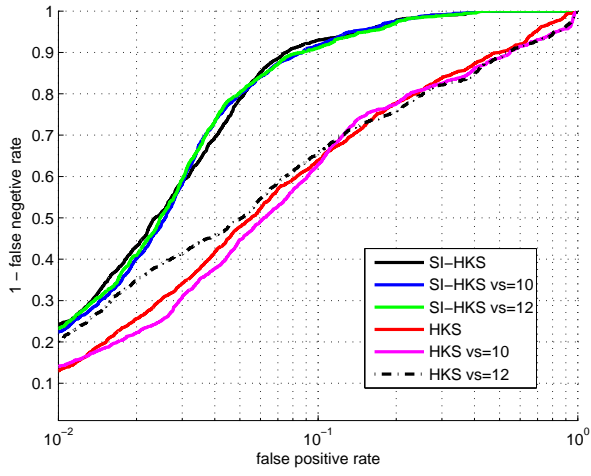
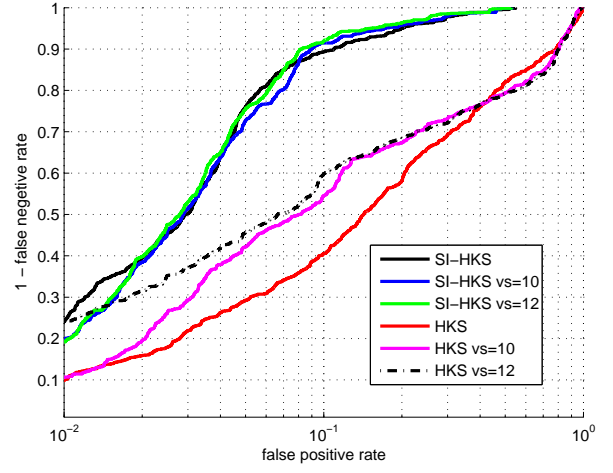
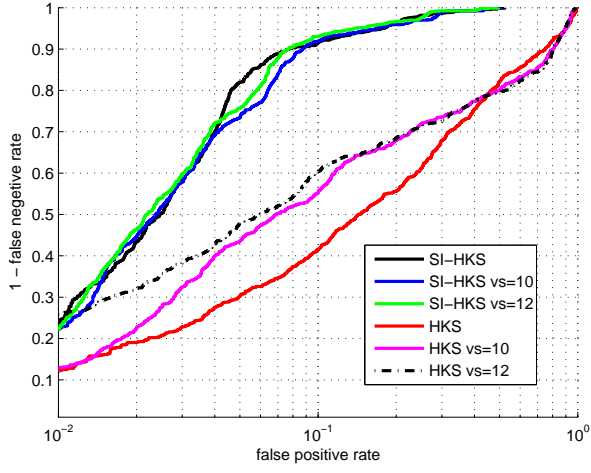


Figure 8: ROC curves of different regions descriptors (“vs” stands for vocabulary size). The following detectors were used (left-to-right, top-to-bottom): vertex weight  $h_t(v, v)$ , edge weight  $1/h_t(v_1, v_2)$ , edge weight  $|h_t(v_1, v_1) - h_t(v_2, v_2)|$ , and edge weight  $1/c(v_1, v_2)$ .

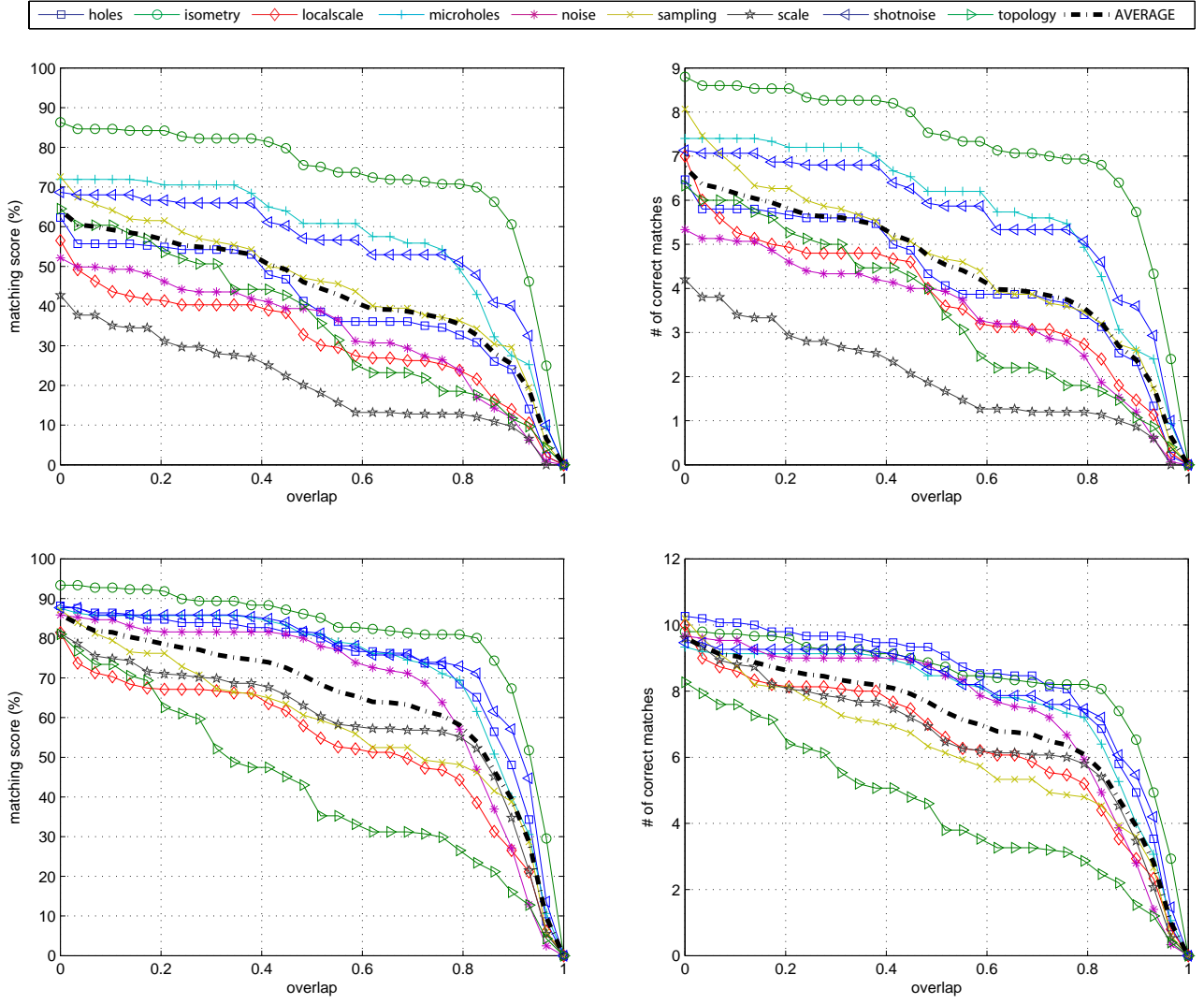


Figure 9: Performance of region descriptors with regions detected using the vertex weight  $h_t(v, v)$ ,  $t = 2048$ . Shown are the HKS descriptor (first row) and SI-HKS descriptor (second row).



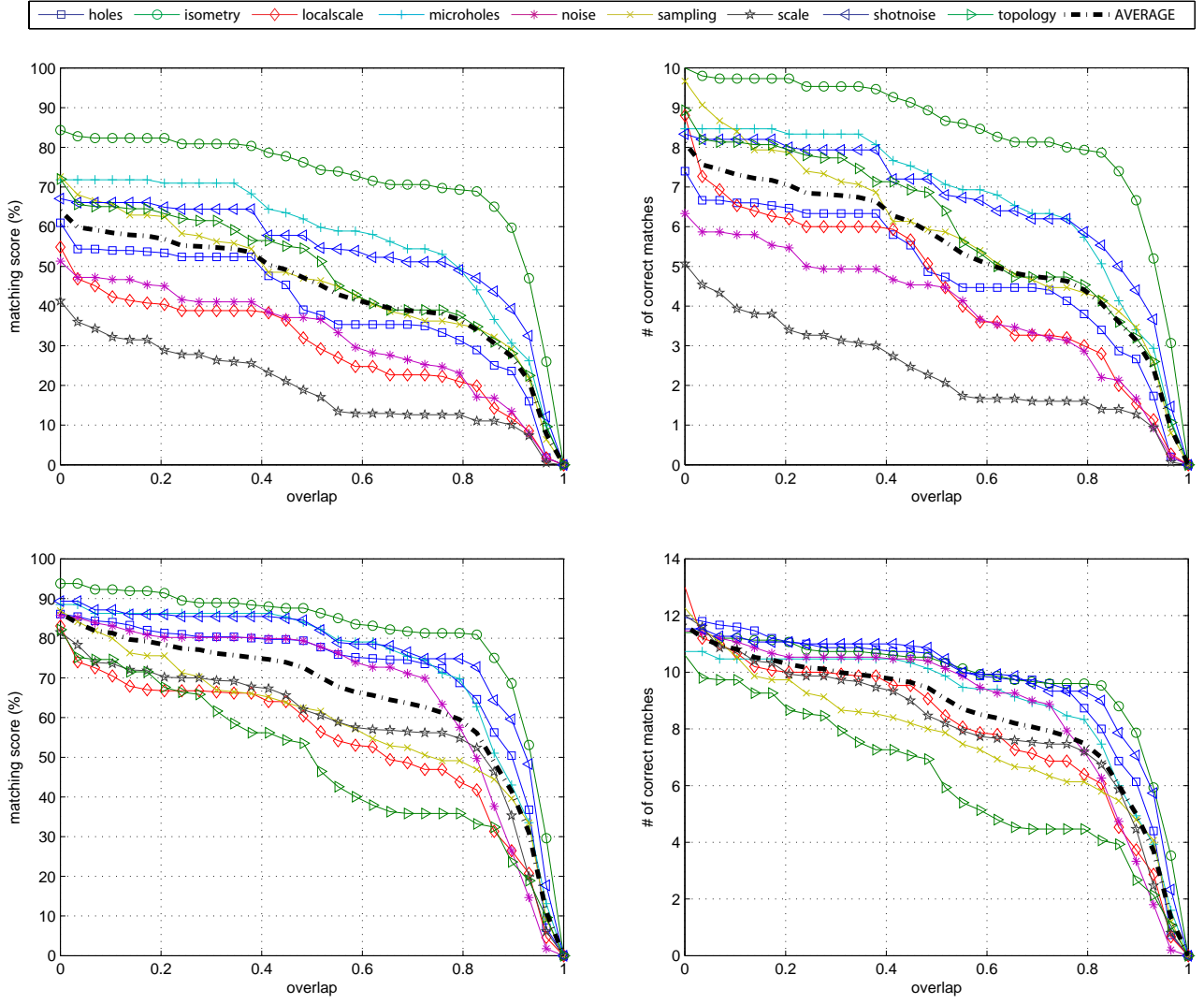


Figure 10: Performance of region descriptors with regions detected using the edge weight  $1/h_t(v_1, v_2)$ ,  $t = 2048$ . Shown are the HKS descriptor (first row) and SI-HKS descriptor (second row).

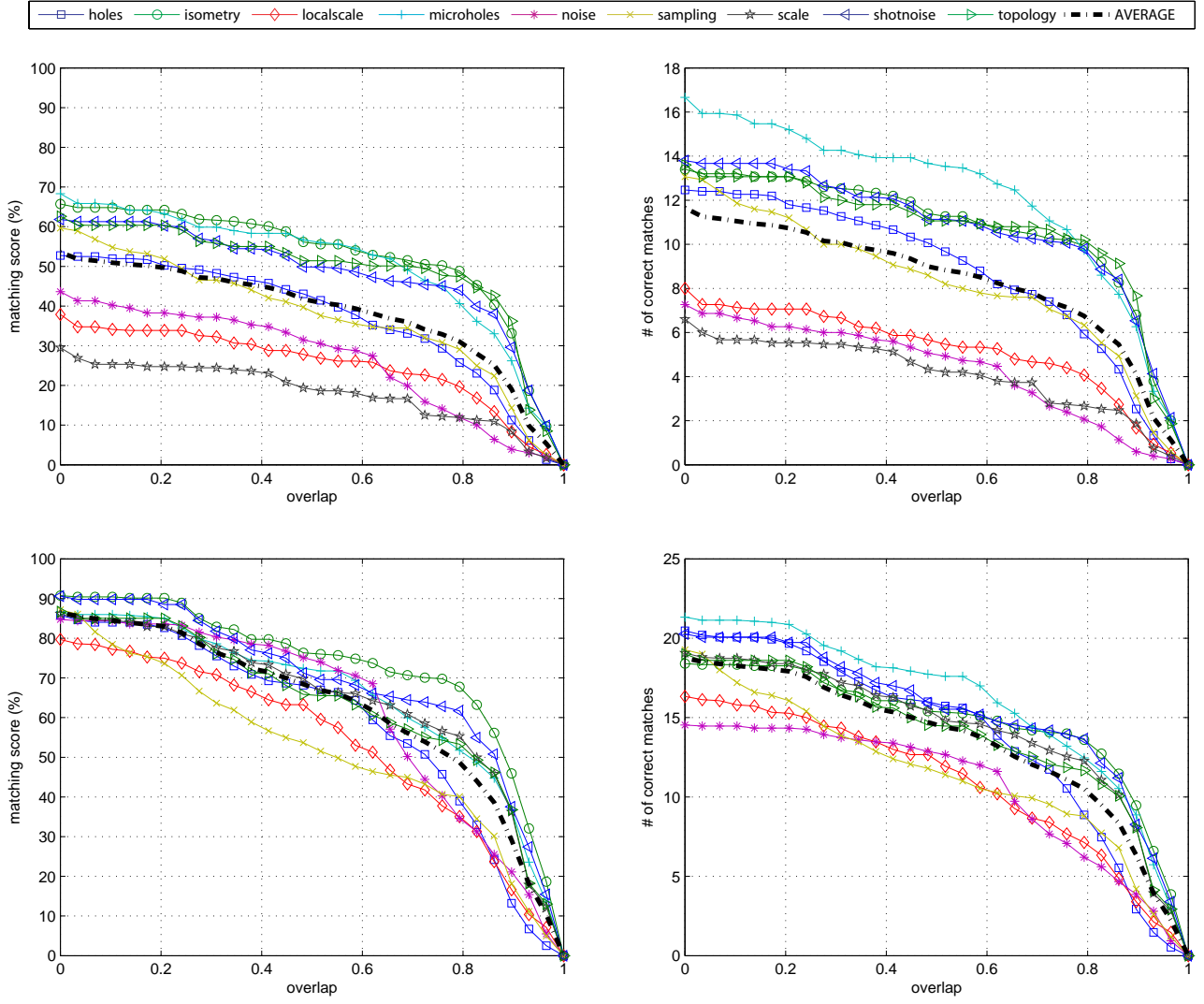


Figure 11: Performance of region descriptors with regions detected using the edge weight  $|h_t(v_1, v_1) - h_t(v_2, v_2)|$ ,  $t = 2048$ . Shown are the HKS descriptor (first row) and SI-HKS descriptor (second row).

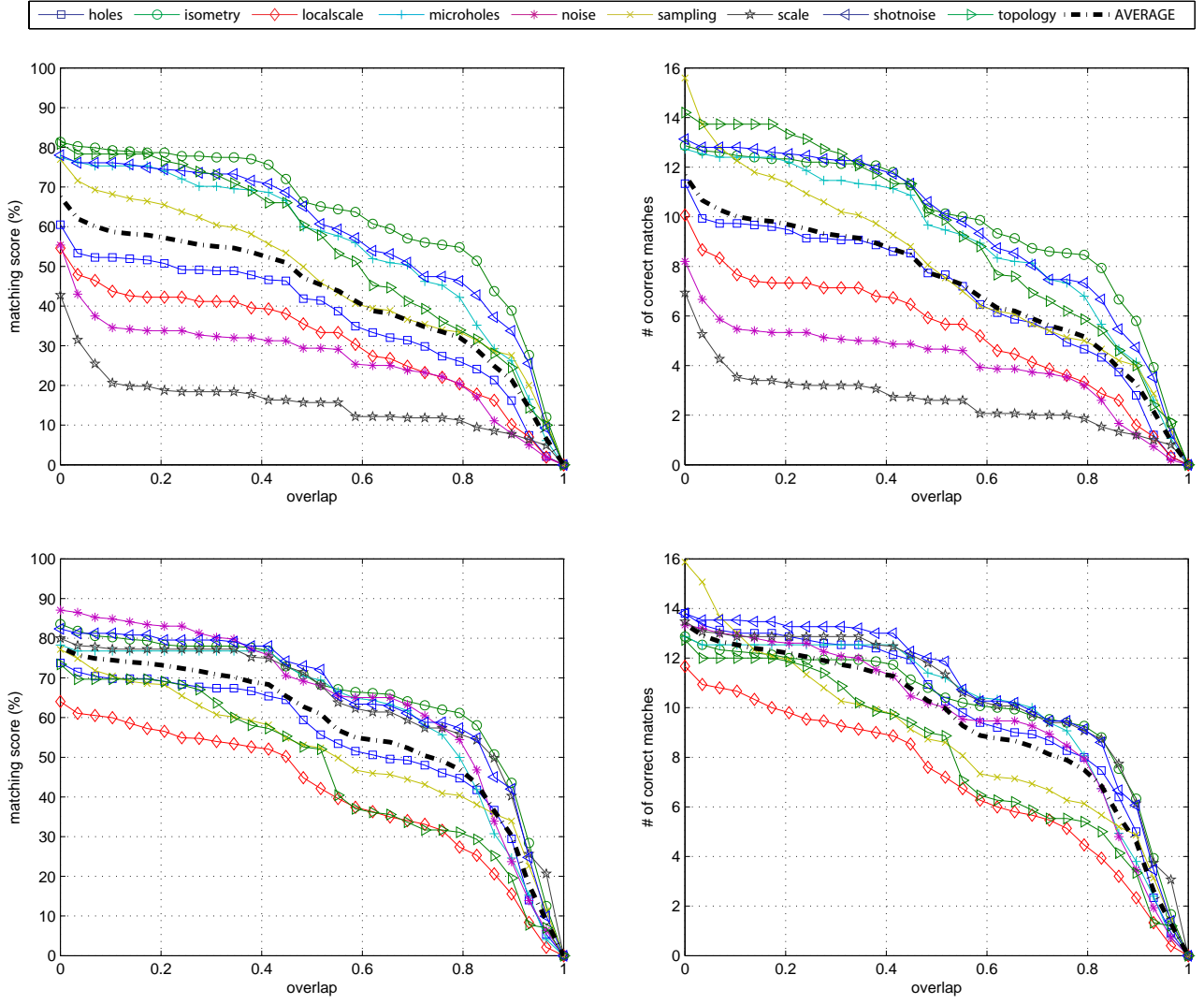


Figure 12: Performance of region descriptors with regions detected using the edge weight  $1/c(v_1, v_2)$ . Shown are the HKS descriptor (first row) and SI-HKS descriptor (second row).

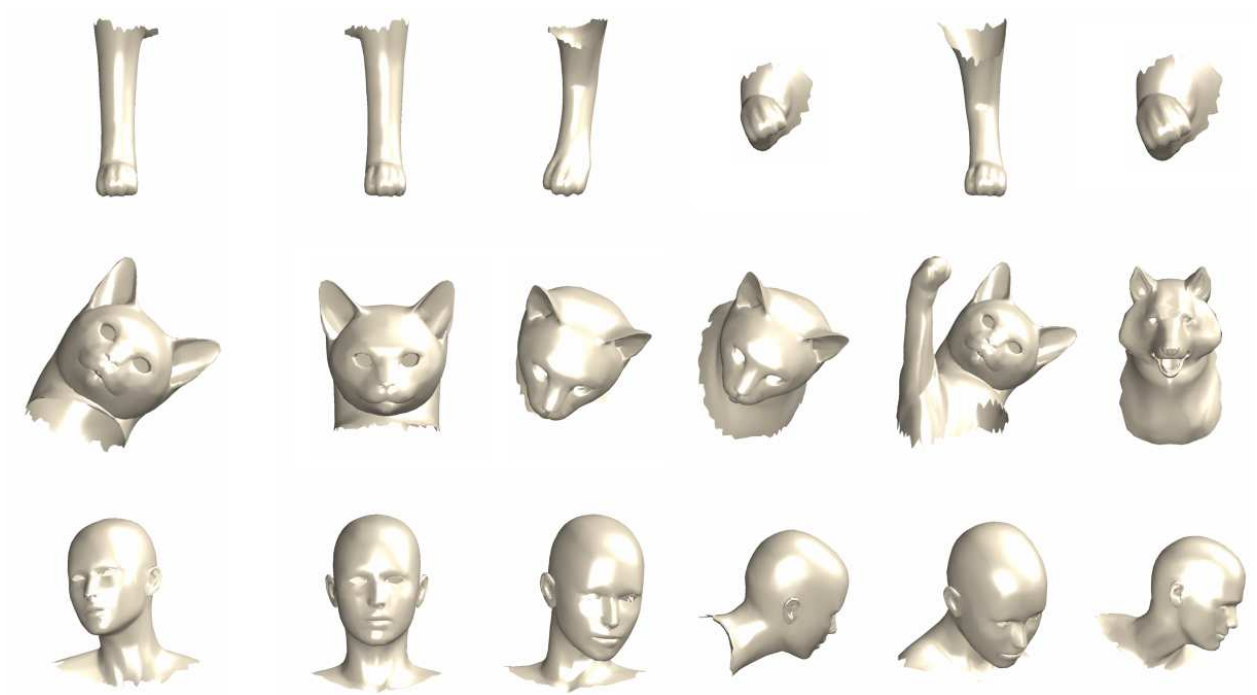


Figure 13: Examples of closest matches found for different query regions on the TOSCA dataset. Shown from left to right are: query, 1st, 2nd, 4th, 10th, and 14th matches. Vertex weight  $h_t(v, v)$  with  $t = 2048$  was used as the detector; average SIHKS was used as the descriptor.