



**HAL**  
open science

## Matching 2D and 3D Articulated Shapes using Eccentricity

Adrian Ion, Nicole M. Artner, Gabriel Peyré, Walter G. Kropatsch, Laurent  
D. Cohen

► **To cite this version:**

Adrian Ion, Nicole M. Artner, Gabriel Peyré, Walter G. Kropatsch, Laurent D. Cohen. Matching 2D and 3D Articulated Shapes using Eccentricity. *Computer Vision and Image Understanding*, 2011, 115 (6), pp.817-834. 10.1016/j.cviu.2011.02.006 . hal-00365019v2

**HAL Id: hal-00365019**

**<https://hal.science/hal-00365019v2>**

Submitted on 25 Mar 2011

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Matching 2D and 3D Articulated Shapes Using the Eccentricity Transform

Adrian Ion<sup>\*,a,b</sup>, Nicole M. Artner<sup>b,c</sup>, Gabriel Peyré<sup>d</sup>, Walter G. Kropatsch<sup>b</sup>,  
Laurent D. Cohen<sup>d</sup>

<sup>a</sup>*INS, University of Bonn, Wegelerstr. 6, 53113, Bonn, Germany*

<sup>b</sup>*PRIP, Vienna University of Technology, Favoritenstr. 9/1863, Vienna, Austria*

<sup>c</sup>*AIT Austrian Institute of Technology, Vienna, Austria*

<sup>d</sup>*CEREMADE, UMR CNRS 7534, Université Paris-Dauphine, Place du Maréchal De  
Lattre De Tassigny, 75775 Paris Cedex 16, France*

---

## Abstract

This paper presents a novel method for 2D and 3D shape matching that is insensitive to articulation. It uses the eccentricity transform, which is based on the computation of geodesic distances. Geodesic distances computed over a 2D or 3D shape are articulation insensitive. The eccentricity transform considers the length of the longest geodesics. Histograms of the eccentricity transform characterize the compactness of a shape, in a way insensitive to rotation, scaling, and articulation. To characterize the structure of a shape, a histogram of the connected components of the level sets of the transform is used. These two histograms make up a highly compact descriptor and the resulting method for shape matching is straightforward. Experimental results on established 2D and 3D benchmarks show results similar to more complex state of the art methods, especially when considering articulation. The connection between the geometrical modification of a shape and the corresponding impact on its histogram representation is explained. The influence of the number of bins in the two histograms and the respective importance of each histogram is studied in detail.

---

\*Corresponding author.

*Email addresses:* [ion@ins.uni-bonn.de](mailto:ion@ins.uni-bonn.de) (Adrian Ion), [artner@prip.tuwien.ac.at](mailto:artner@prip.tuwien.ac.at) (Nicole M. Artner), [peyre@ceremade.dauphine.fr](mailto:peyre@ceremade.dauphine.fr) (Gabriel Peyré), [krw@prip.tuwien.ac.at](mailto:krw@prip.tuwien.ac.at) (Walter G. Kropatsch), [cohen@ceremade.dauphine.fr](mailto:cohen@ceremade.dauphine.fr) (Laurent D. Cohen)

*Key words:* eccentricity transform, shape matching, articulation, geodesic distance

---

## 1. Introduction

The recent increase in available 3D models and acquisition systems has created the need for efficient retrieval of stored models, making 3D shape matching gain attention also outside the computer vision community. Together with its 2D counterpart, 3D shape matching is useful for identification and retrieval in classical vision tasks, but can also be found in Computer Aided Design/Computer Aided Manufacturing (CAD/CAM), virtual reality (VR), medicine, molecular biology, security, and entertainment [1].

Shape matching requires to set up a signature that characterizes the properties of interest for the recognition [2]. Depending on the task, the invariance of this signature to local deformations such as articulation is important for the identification of 2D and 3D shapes. Matching can then be carried out over this (usually lower dimensional) space of signatures.

### 1.1. Related work

Most shape descriptors are computed over a transformed domain that amplifies the important features of the shape while throwing away ambiguities such as translation, rotation or local deformations.

For 2D shapes, the *Fourier transform of the boundary curve* [3] is an example of such a transformed-domain descriptor adapted to smooth shapes. *Shape transformations computed with geodesic distances* [4, 5] lead to signatures invariant to isometric deformations such as bending or articulation. To capture salient features of 2D shapes, local quantities such as *curvature* [6] or *shape contexts* [7] can be computed. They can be extended to *bending invariant signatures* using geodesic distances [8, 9]. Another possibility is to represent a shape as a collection of modestly overlapping disk components, which contain both local geometric and structural information [10]. More global features include the *Laplace spectra* [11] and the *skeleton* [12]. *Contour flexibility* [13] is a novel shape descriptor of planar contours, which obtains both local and global features from a contour. It represents the deformable potential at each point along a contour. The *rolling penetrate descriptor* [14] combines the advantages of contour-based and region-based methods, and provides a unified scheme to handle various shapes, geometrical transforms, noise, distortion and occlusion. Some transformations involve

the computation of a function defined on the shape, for instance the solution to a linear *partial differential equation* [15] or *geometric quantities* [16].

Among approaches matching 3D shapes, existing methods can be divided into [1]: *statistical* descriptors, like for example geometric *3D moments* employed by [17, 18], 3D moment invariants [19], and the *shape distribution* [16, 20]. A novel shape representation based on [19] is *3D gray level moment invariants* [21], which are independent of translation, scaling and rotation. *Extension-based* descriptors are calculated from features sampled along certain directions from a position within the shape [22, 23]. *Volume-based* descriptors use the volumetric representation of a 3D shape to extract features (examples are *Shape histograms* [24], *Model Voxelization* [25], and point set methods [26]). The *3D shape impact descriptor* [27] considers a 3D object as a distributed 3D mass and it is indirectly computed from the resulting fields. The field is described using both Newton’s and general relativity laws. In [28], 3D shapes are sampled using a technique based on critical points of the eigenfunctions of the *Laplace-Beltrami operator*. A point-based statistical descriptor is used that incorporates an approximation of the *geodesic shape distribution* and other geometric information describing the surface at that point. Matching is carried out using Bipartite graph matching. Descriptors using the *surface geometry* compute curvature measures and/or the distribution of surface normal vectors [29, 30]. *Image-based* descriptors reduce the problem of 3D shape matching to an image similarity problem by comparing 2D projections of the 3D shapes [31, 32, 33]. *Reeb* graphs have been used to match the topology of two shapes [34, 35]. *Skeletons* are intuitive shape descriptions and can be obtained from a 3D shape by applying a thinning algorithm on the voxelization of a solid object like in [36]. Descriptors using *spin images* work with a set of 2D histograms of the shape geometry and a search for point-to-point correspondences is done to match 3D objects [37]. In [38], 3D shapes are automatically decomposed into parts using *topological features of the Laplace-Beltrami Eigenfunctions*. The parts of near isometric shapes are registered to each other.

A generic class of approaches for 2D and 3D shape retrieval is to define a metric between pairs of points on the shape, and compare either directly these metric spaces or compare features extracted from these spaces. An important goal of this metric space design is to make the shape retrieval more or less invariant to bending and articulations. The Gromov-Hausdorff framework directly compares metric spaces [39], and can be used for shape retrieval in conjunction with geodesic metric spaces [4] or diffusion spaces that are more

robust to topological noise [40, 41]. Associated to the diffusion metric, it is possible to define a metric using a dimensionality reduction within a few eigenvectors of the Laplacian [42]. To speed-up retrieval applications, one can consider low dimensional features extracted from these metric spaces. This can be for instance the set of geodesic distances between critical points of the Laplace eigenfunction [28], local distributions of geodesic curves [8], statistical moments of a diffusion distance [15]. A popular class of approaches considers histograms, such as the distribution of Euclidean distances [16], of the mean geodesic distance [5] or the maximum geodesic distance [43, 44], or bags of features [45]. This article elaborates on this idea of building compact geodesic descriptors using two different kinds of histograms to represent faithfully the distribution of geodesic distances.

Considering the underlying transform on which our descriptor is built, probably the most similar works in shape matching are [5, 34], where instead of the length of the geodesic to the point furthest away (this work), the mean or the sum over all points are considered. Distance based transforms are also used in [15, 12, 46], where the length of random or shortest paths to an (existing) boundary are computed. Describing a shape as a normalized histogram of the values of a function at all points of the shape, as in the case of the normalized *ECC histogram* in this paper, is conceptually similar to the shape distributions in [16], where different functions and histogram comparison methods are considered.

In [8] a *model of articulated objects* is presented. It is defined as a union of (rigid) parts  $\mathcal{O}_i$  and joints (named ‘junctions’ by the authors). An articulation is defined as a transformation that is rigid when limited to any part  $\mathcal{O}_i$ , but can be non-rigid on the junctions. An articulated instance of an object is an articulated object itself (actually the same object) that can be articulated back to the original one. The term *articulated shape* refers to the shape of an articulated object in a certain pose. In the context of shape matching the concept of *articulated shape* means that shapes that belong to articulations of the same object, belong to the same class. Assuming that the size of the junctions is very small compared to the size of the parts  $\mathcal{O}_i$ , it is shown that the variation of the geodesic distance<sup>1</sup> during articulation is small and that geodesic distances are articulation insensitive.

Similar to the method in [8], our method does not explicitly involve any

---

<sup>1</sup>Called ‘inner-distance’ in [8].

part models. In [8] the part based articulation model was used to support the analysis of the properties of the geodesic distance under articulation. We have briefly recalled it in Section 1.1 to aid the definition of an articulated shape. Finding the correspondences between all the parts of two shapes is an *NP*-complete problem in graph theory (known also as the ‘matching’ of two graphs) and requires the correct decomposition of the unknown object into parts. A one-to-one correspondence (bijection) for all parts is not always possible as some parts might be missing (e.g. due to segmentation errors).

### 1.2. Contribution

The eccentricity transform of a shape associates to each of its points the distance to the point furthest away. It is based on the computation of geodesic distances and thus robust with respect to articulation. It is robust against minor segmentation errors, and Salt and Pepper like noise [47], and stable in the presence of holes (i.e. it is defined in the same way for shapes with and without holes, and does not require pre-selection of for e.g. a single closed boundary to be processed).

In a common framework, we propose histograms built on the eccentricity transform as descriptors for 2D and 3D shape matching. The descriptor consists of two histograms: the *ECC histogram*  $\mathbf{h}$  which is the normalized histogram of the eccentricity transform of the shape, and the *ECC structure histogram*  $\mathbf{s}$  which is the histogram of the number of connected components of the level sets of the eccentricity transform. The descriptor is invariant to changes in orientation, scale, and articulation. It requires only a simple representation and can be efficiently matched. We present an in-depth study of the properties of the approach (relation between shape and descriptor, parameters), supported by experimental results, and an analysis of the results and possibilities for improvement. The descriptor is computed on four different domains using geodesic distances: inside 2D shapes, inside the volume, inside the border voxels, or on the surface meshes of 3D shapes, and compared to state of the art methods. These numerical results support that the proposed approach performs similarly and in some case better than the state of the art on databases of articulated 2D and 3D shapes. Initial results using only the ECC histogram  $\mathbf{h}$  have been presented for 2D in [43], and for 3D (volumetric representation only) in [44].

To the best of our knowledge, this is the first approach applying the eccentricity transform to the problem of shape matching.

### 1.3. Overview of the Paper

The paper is organized as follows: Section 2 recalls the eccentricity transform and discusses used variants and computation. Section 3 explains the proposed matching method and discusses pros and cons of the descriptor (Section 3.4). In Section 4 experiments, and the effect of the parameters are presented, followed by future work (Section 5) and conclusion (Section 6). The Appendix recalls the algorithm used to compute the eccentricity transform.

## 2. Eccentricity Transform

The following definitions and properties follow [47, 48], and are extended to  $n$ -dimensional domains.

Let the shape  $\mathcal{S}$  be a closed set in  $\mathbb{R}^n$ . A path  $\pi$  in  $\mathcal{S}$  is the continuous mapping from the interval  $[0, 1]$  to  $\mathcal{S}$ . Let  $\Pi(\mathbf{p}_1, \mathbf{p}_2)$  be the set of all paths between two points  $\mathbf{p}_1, \mathbf{p}_2 \in \mathcal{S}$  within the set  $\mathcal{S}$ .

The *geodesic distance*  $d(\mathbf{p}_1, \mathbf{p}_2)$  between two points  $\mathbf{p}_1, \mathbf{p}_2 \in \mathcal{S}$  is defined as the length  $\lambda(\pi)$  of the shortest path  $\pi \in \Pi(\mathbf{p}_1, \mathbf{p}_2)$

$$d(\mathbf{p}_1, \mathbf{p}_2) = \min\{\lambda(\pi) \mid \pi \in \Pi(\mathbf{p}_1, \mathbf{p}_2)\}, \quad (1)$$

where the length  $\lambda(\pi)$  is

$$\lambda(\pi(t)) = \int_0^1 \|\dot{\pi}(t)\| dt,$$

$\pi(t)$  is a parametrization of the path from  $\mathbf{p}_1 = \pi(0)$  to  $\mathbf{p}_2 = \pi(1)$ ,  $\dot{\pi}(t)$  is the derivative of the curve with respect to  $t$ , and  $\|\cdot\|$  denotes the  $L_2$ -norm.

Any path  $\nu \in \Pi(\mathbf{p}_1, \mathbf{p}_2)$ , satisfying  $\lambda(\nu) = d(\mathbf{p}_1, \mathbf{p}_2)$  is called a geodesic (path).

The *eccentricity transform* of  $\mathcal{S}$  is defined as,  $\forall \mathbf{p} \in \mathcal{S}$

$$ECC(\mathcal{S}, \mathbf{p}) = \max\{d(\mathbf{p}, \mathbf{q}) \mid \mathbf{q} \in \mathcal{S}\}. \quad (2)$$

To each point  $\mathbf{p}$  it assigns the length of the geodesic path(s) to the points farthest away from it.

The definition above accommodates  $n$ -dimensional objects embedded in  $\mathbb{R}^n$  as well as  $n$ -dimensional objects embedded in higher dimensional spaces

Table 1: Types of manifolds used for matching.

Name	input	computing on	$\mathcal{S}$ ( $d_\varepsilon$ is used)
<b>ECCobj2D</b>	2D	2D: whole shape	4-connected binary 2D shape
<b>ECCobj</b>	3D	3D: whole shape	6-connected 3D voxel shape
<b>ECCborder</b>	3D	3D: border voxels	6 connected voxel surface in 3D, made out of voxels of the shape that are 26 connected to a background voxel
<b>ECCmesh</b>	3D	2D: triangular mesh	connected triangular mesh of the surface of the 3D shape


(e.g. the 2D manifold given by the surface of a closed 3D object). The distance between any two points whose connecting segment is contained in  $\mathcal{S}$ , is computed using the  $L_2$ -norm, i.e. distances are not computed on a graph, but are a discretization of the continuous geodesic distance. For a definition of the *ECC* of a graph see [47].

The *ECC* is quasi-invariant to articulated motion and robust against salt and pepper noise, which creates small (typically 1 pixel) holes in the shape [47]. An analysis of the variation of geodesic distance under articulation can be found in [8].

An *eccentric point* is a point  $\mathbf{q}$  that reaches a maximum in Equation 2, and for most shapes, all eccentric points lie on the border of  $\mathcal{S}$  [47]. An *eccentric path* of a point  $\mathbf{p}$  is a geodesic to one of its eccentric points. The (*geodesic*) *center* is the set of points that have the smallest eccentricity (global minimum). The *diameter* of a shape  $\mathcal{S}$  is the maximum *ECC*, which is the length of the longest geodesic path in  $\mathcal{S}$ .

The classes of  $2n$ -connected discrete shapes  $\mathcal{S}$  defined by points on a square grid  $\mathbb{Z}^n$ ,  $n \in \{2, 3\}$ , as well as connected triangular meshes representing the surface of the 6-connected 3D shapes are considered in the paper. Table 1 shows the types of manifolds used in this article, for which *ECC* is computed. For ECCobj2D, ECCobj, and ECCborder, paths need to be contained in the area of  $\mathbb{R}^n$  defined by the union of the support squares/cubes for the pixels/voxels of  $\mathcal{S}$ . For ECCmesh, paths need to be contained in the 2D manifold defined by the union of the triangles of the mesh (including the interior of the triangles). The used (approximated) metric is in all cases the Euclidean based geodesic distance  $d_\varepsilon$ , i.e. the distance between the endpoints of any line segment included in the shape is computed using the Euclidean distance. When the resolution of the shapes increases ECCborder





measure	transf.	shape							
		2	3	4	5	6	7	8	9
mean pixel diff.	DT	0.00	0.00	0.01	0.01	0.03	0.03	0.19	0.07
	GGF	0.02	0.05	0.12	0.20	0.35	0.02	0.02	0.03
	ECC	0.00	0.01	0.09	0.26	0.48	0.02	0.02	0.03
max pixel diff.	DT	0.20	0.20	0.21	0.24	0.39	0.12	0.99	1.00
	GGF	0.07	0.11	0.23	0.31	0.54	0.10	0.07	0.09
	ECC	0.06	0.13	0.32	0.67	0.76	0.11	0.06	0.17
% same to 1 <sup>st</sup>	DT	98%	95%	90%	85%	68%	46%	30%	80%
	GGF	1%	0%	0%	0%	0%	0%	1%	1%
	ECC	88%	76%	41%	22%	8%	11%	0%	1%
% same to prev.	DT	98%	97%	95%	96%	86%			
	GGF	1%	0%	0%	0%	0%			
	ECC	88%	76%	43%	34%	27%			

Figure 1: Top: original hand followed by 8 modifications: finger removal (2-6), boundary noise (7)(similar to [49]), random missing pixels (8), and adding a large hole (9). Bottom: variation of the values of DT, GGF, and ECC due to the 8 modifications. The *mean* and *max* values give the mean/max difference in the values of pixels in the original and altered shapes, normalized by the difference between the highest and lowest value on the original shape. The “% same” values give for cases (2-6) the ratio of pixels in the altered shape with values unchanged as compared to the whole hand and to the values in the previous shape (1-5), respectively.

and ECCmesh converge to the same value.

### 2.1. Similarity to other transforms

The eccentricity transform is part of a greater class of distance-based transforms along with:

- the *distance transform* (DT) [50] defined as:

$$DT(\mathcal{S}, \mathbf{p}) = \min\{d(\mathbf{p}, \mathbf{q}) \mid \mathbf{q} \in \mathcal{Y} \subseteq \mathcal{S}\}$$

where the set  $\mathcal{Y}$  is called *marker set* and is usually taken as  $\mathcal{Y} = \partial\mathcal{S}$ ;

- the *global geodesic function* (GGF) [34] defined as:

$$GGF(\mathcal{S}, \mathbf{p}) = \int_{\mathbf{q} \in \mathcal{S}} d(\mathbf{p}, \mathbf{q}).$$

*Shortest* geodesics (DT) have the advantage that they locally characterize the shape and are invariant to deformations in all other parts of the shape. On the other side, the shorter a path, the higher the perturbation that is created by a hole (obstacle).

ECC uses *longest* geodesics and has thus the highest stability with respect to small holes. With respect to deformations, ECC is invariant to non-isometric deformations that do not affect the parts where eccentric points lie, i.e. as long as a certain part does not contain any eccentric points it can even be removed, without changing the values for the remaining points of the shape [51] (e.g. for a shape like a “T” with the vertical line connected to the center of an at least twice longer horizontal line, removing the vertical line will have no effect on the eccentricity values on the horizontal line).

Figure 1 shows the changes in the mean and maximum values of the three transforms when the shape of a hand from the Kimia 99 database [52] is changed. The DT is more local and thus insensitive to the missing fingers, but comparable in the case of boundary noise, and the highest variation with respect to missing points inside. Due to the averaging that the GGF does, it has a maximum variation which is in all cases smaller than that of the ECC, but almost all pixels change even for a minor change in the shapes. The variation of the ECC is larger than the one of the GGF, but with a considerable amount of pixels keeping the same value before and after a finger is removed.

Both the DT and the GGF have previously been used as a basis for shape matching methods. The DT is closely related to the skeleton/medial axis/medial surface [46, 53] of a 2D/3D shape, on which shock graphs [12] are built. Matching of shapes can be carried out by matching their shock graphs or medial surfaces. A continuous definition of the GGF has been used in [34] to match the topology of 3D shapes based on the similarity of their Reeb graphs. In [5] a discrete version of the squared GGF is used to build so called “geodesic shape distributions”. Kernel density estimation is used to get a smooth function from the computed samples, and the Jensen-Shannon divergence is used to match two distributions. Our normalized eccentricity histograms are in spirit similar to the geodesic shape distributions. Differences lie in the used function (ECC vs squared GGF) and in the normalization and matching of histograms ( $L_2$ -norm). In addition we use a second histogram to characterize the structure of a shape.

## 2.2. Computation

Furthest point computation has been approached in [54], where an algorithm is presented which finds for each vertex of a simple polygon the vertex that is furthest away (eccentric). Later, [55, 56] proposed an efficient algorithm for simply connected shapes on the hexagonal and dodecagonal grid. The concept of the eccentricity of a vertex can be found in classical graph theory books [57, 58], and the concept of the eccentricity transform<sup>2</sup> in recent discrete geometry [59] and mathematical morphology books [50]. Computation is not discussed and no references to holes in a shape are made.

The straightforward approach to compute  $ECC(\mathcal{S})$  is: for each point of  $\mathcal{S}$ , compute the distance to all other points and take the maximum. A faster computation and efficient approximation algorithms are presented in [48]. For this paper the fastest one, algorithm ECC06, is used.

ECC06 relies on the computation of the *shape bounded single source distance transform*<sup>3</sup>  $D^{\mathcal{S}}(\mathbf{p})$  (Figure 2(b)), which is computed for estimated eccentric point candidates in an iterative manner (see the Appendix for more details).  $D^{\mathcal{S}}(\mathbf{p})$  associates to each point  $\mathbf{q} \in \mathcal{S}$  the geodesic distance to  $\mathbf{p}$ .  $D^{\mathcal{S}}$  can be computed using Fast Marching [60] (FM), without the need to explicitly build a neighborhood graph. The runtime complexity of FM is  $O(N \log(N))$  steps, for  $N = |\mathcal{S}|$  grid points, where  $|\cdot|$  denotes set cardinality.

The complexity for computing  $ECC(\mathcal{S})$  using ECC06 and Fast Marching is  $O(KN \log(N))$ , where  $2 \leq K \leq |\partial\mathcal{S}|$ ,  $K \in \mathbb{N}$  depends on the shape and is the number of eccentric point candidates that are evaluated.

Figure 2 shows a comparison of the geodesic and Euclidean distances. Figures 3 and 4 show the eccentricity transform of a 2D, respectively 3D, shape. For the 3D shape, the eccentricity transform is presented for the whole shape (ECCobj), for the border voxels (ECCborder), and the surface mesh (ECCmesh). Figure 5 shows the difference between ECCobj and ECCborder, both using distances computed on a voxel description of  $\mathcal{S}$ .

## 3. Eccentricity Histogram Matching

To match two shapes we first create a shape descriptor for each of them and then match these descriptors to obtain a similarity measure. The pro-

---

<sup>2</sup>Known in the mathematical morphology community as the *propagation function*.

<sup>3</sup>Also called geodesic distance function with marker set  $\mathbf{p}$ .

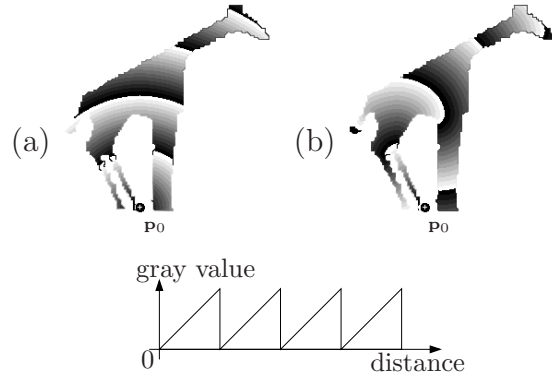


Figure 2: Euclidean (a) and geodesic (b) distance function, for starting point  $p_0$ . Gray values are distances modulo a constant.



Figure 3: *ECC* of example binary shape (point with smallest *ECC* marked).

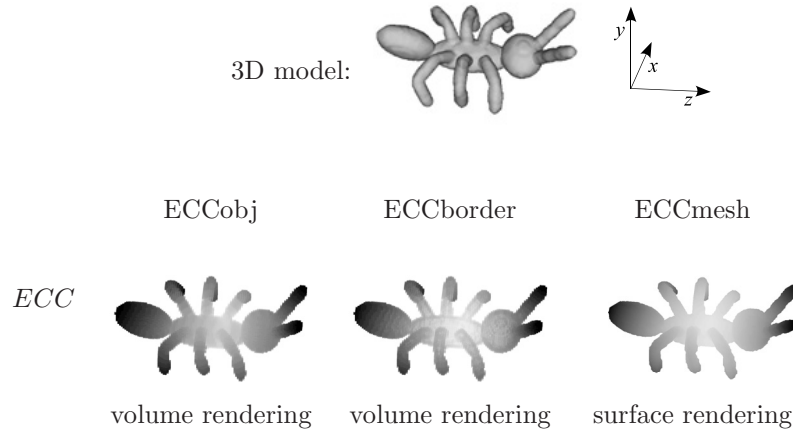


Figure 4: Top: 3D model of an ant. Bottom: *ECCobj*, *ECCborder*, *ECCmesh* (darker = higher *ECC* value). Notice that in all three cases, the transform has its minimum in the body of the ant (center) and the values get larger as going to the extremities.

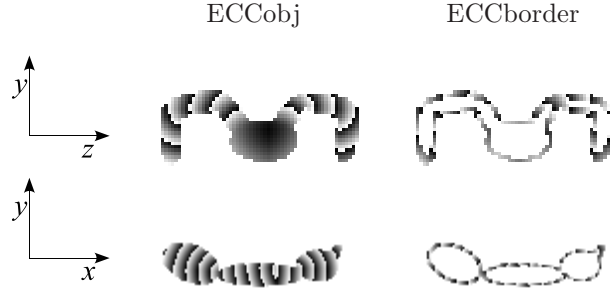


Figure 5: Comparison between the two computations of  $ECC$  on voxels:  $ECC_{obj}$  and  $ECC_{border}$ . The Figure shows two vertical cuts through the ant in Figure 4.

posed shape descriptor is made out of two components: the first one characterizes the compactness (geometry) of the shape (Section 3.1) and the second one its structure (Section 3.2). The descriptor is highly compact, which is an advantage for real time retrieving and low memory devices, it is invariant under many natural deformations, it can handle shapes without as well as with holes<sup>4</sup> (Figure 8 (g) and (h)), and gives good results, comparable to the presented state of the art methods (experiments follow in Section 4).

### 3.1. $ECC$ histogram

The first component the shape descriptor is the histogram  $\mathbf{h}$  of the eccentricity transform  $ECC$  of the shape  $\mathcal{S}$ . We use  $k_h$  bins for the histogram. The eccentricity histogram is the vector  $\mathbf{h} \in \mathbb{R}^{k_h}$  defined by:  $\forall i = 1, \dots, k_h$

$$\mathbf{h}(\mathcal{S}, i) = \frac{1}{|\mathcal{S}|} \# \left\{ \mathbf{p} \in \mathcal{S} \mid \frac{i-1}{k_h} \leq \frac{ECC(\mathcal{S}, \mathbf{p}) - m}{M - m} < \frac{i}{k_h} \right\},$$

where  $|\mathcal{S}|$  is the number of pixels/voxels/vertices in  $\mathcal{S}$ , and  $m$  and  $M$  are the smallest, respectively, largest eccentricity values over  $\mathcal{S}$ . A discussion about choosing the number of bins  $k_h$  follows in Section 4.4. The obtained histogram only contains bins for the values which exist in the eccentricity transform, i.e. from minimum to maximum eccentricity, and the sum over all bins is 1. Figures 6 and 7 show examples of eccentricity histograms for 2D and 3D shapes with different geometric features. Note that the histogram  $\mathbf{h}$  is invariant under Euclidean transformations, scaling and isometric bending of  $\mathcal{S}$  (Figure 8 shows examples).

---

<sup>4</sup>As opposed to methods selecting a single boundary, usually the outer/longest one, which cannot differentiate for example a disk from a 2D torus.

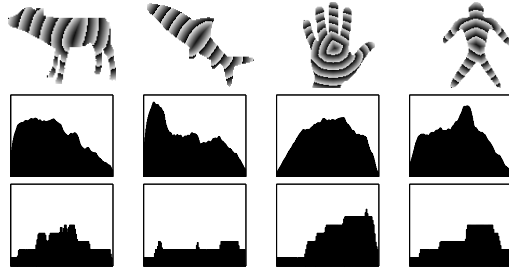


Figure 6: Top: ECCObj2D for some 2D shapes. Middle and bottom: corresponding histograms  $\mathbf{h}$  respectively  $\mathbf{s}$ .

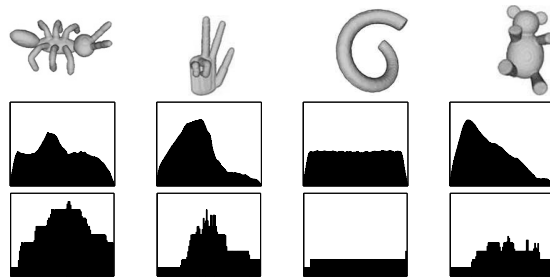


Figure 7: Top: example 3D shapes. Middle and bottom: corresponding ECCObj histograms  $\mathbf{h}$  respectively  $\mathbf{s}$ .

### 3.2. ECC structure histogram

The second component of our shape descriptor is a histogram of the number of connected components of the discrete level-sets of the eccentricity transform. A discrete level-set of  $ECC(\mathcal{S})$  is the set of points of the shape having their eccentricity value in a certain domain  $(a, b]$ . More formally:

$$\mathcal{L}(a, b) = \{\mathbf{p} \in \mathcal{S} \mid a < ECC(\mathcal{S}, \mathbf{p}) \leq b\}.$$

The eccentricity structure histogram is the vector  $\mathbf{s} \in \mathbb{R}^{k_s}$  defined by:  $\forall i = 1, \dots, k_s$

$$\mathbf{s}(\mathcal{S}, i) = \#_C \{\mathcal{L}(e_{i-1}, e_i)\},$$

where  $\#_C$  denotes the number of connected components. We use 8 connectivity in 2D, 26 connectivity for the 3D voxel representations, and vertex adjacency for the triangle meshes. For a function which is based on Euclidean distances, to obtain at least 8 respectively 26 connected isolines/isosurfaces, the thresholds  $e_i$  have to satisfy that  $e_i - e_{i-1} \geq g$  [61]. Where  $g = 1$  is the distance between two closest grid points. For the case of the triangle meshes, we use  $g = \sqrt{2}$ , which is the length of the longest edge in our meshes. If  $e_i - e_{i-1} < g$  then the pixels covering the points of the same (connected) continuous isoline will not be connected in the discrete level-sets. The values for  $e_i$  are computed as:

$$e_i = m + (M - m) \frac{i}{k_s},$$

where as before  $m, M$  are the minimum respectively maximum eccentricity. We take  $e_0 = m - \epsilon$ ,  $\epsilon > 0$  to ensure that the geodesic center is included in the first bin. If  $(M - m)/g < k_s$ , which means that  $e_i - e_{i-1} < g$ , we compute  $\mathbf{s}$  for  $k'_s = \lfloor (M - m)/g \rfloor$  and then resize the obtained histogram to the required  $k_s$  bins using bilinear interpolation.

### 3.3. Comparison of histograms

To match the descriptors of two shapes  $\mathcal{S}$  and  $\tilde{\mathcal{S}}$ , it is necessary to compute the distance between the corresponding histograms. Let  $\mathbf{h}, \mathbf{s}$  and  $\tilde{\mathbf{h}}, \tilde{\mathbf{s}}$  be the two histograms of  $\mathcal{S}$  respectively  $\tilde{\mathcal{S}}$  computed as in Sections 3.1 and 3.2. The distance between two histograms  $\mathbf{h}, \tilde{\mathbf{h}} \in \mathbb{R}^k$  is measured using the  $L_2$ -norm:

$$\delta(\mathbf{h}, \tilde{\mathbf{h}}) \stackrel{\text{def.}}{=} \|\mathbf{h} - \tilde{\mathbf{h}}\|. \quad (3)$$

One could use more elaborate metrics such as the  $\chi^2$  statistic [7] or those defined in [16]. In numerical experiments we found that all these metrics give results similar to  $\delta$ , which is the easiest and fastest to compute (discussion follows in Section 5).

The dissimilarity  $\Delta(\mathcal{S}, \tilde{\mathcal{S}})$  of two shapes  $\mathcal{S}$  and  $\tilde{\mathcal{S}}$  is computed as a weighted distance of their eccentricity histograms  $\mathbf{h}, \tilde{\mathbf{h}}$  and structure histograms  $\mathbf{s}, \tilde{\mathbf{s}}$ :

$$\Delta(\mathcal{S}, \tilde{\mathcal{S}}) \stackrel{\text{def.}}{=} w \cdot \delta(\mathbf{h}, \tilde{\mathbf{h}}) + (1 - w) \cdot \frac{\delta(\mathbf{s}, \tilde{\mathbf{s}})}{\min(\|\mathbf{s}\|, \|\tilde{\mathbf{s}}\|)}, \quad (4)$$

where  $w$  is a mixing parameter.

The histograms  $\mathbf{h} \in \mathbb{R}^{k_h}$  lie in the  $k_h - 1$  dimensional simplex with vertices  $\mathbf{h}_r = (h_1, \dots, h_{k_h})$ ,  $h_i = 1$  for  $i = r$  and 0 otherwise. Thus we have  $0 \leq \delta(\mathbf{h}, \tilde{\mathbf{h}}) \leq \sqrt{2}$  for any  $\mathbf{h}, \tilde{\mathbf{h}} \in \mathbb{R}^{k_h}$ . The values of the bins  $\mathbf{s}(\mathcal{S}, i) \geq 1$  and can be arbitrarily large depending on the number of “parts” of the object. The expression  $\min(\|\mathbf{s}\|, \|\tilde{\mathbf{s}}\|) \geq \sqrt{k_s}$ , and  $\delta(\mathbf{s}, \tilde{\mathbf{s}}) \geq 0$ . For similar shapes (same or similar looking classes) both terms in Equation 4 have values around 0 (in the same range), and extend the descriptive power of the other one. Matching very different shapes (e.g. are horses more similar to scissors than to cars?) is not considered.

In the remainder of the paper, we will use *eccentricity histogram* to denote  $\mathbf{h}$ , *structure histogram* to denote  $\mathbf{s}$ , and *eccentricity based histograms* to refer to both.

### 3.4. Characteristics of the ECC based histograms: from $\mathcal{S}$ to $\mathbf{h}, \mathbf{s}$ and back

The eccentricity histogram  $\mathbf{h}$  characterizes the compactness of the shape (e.g. a flat histogram characterizes a very elongated shape, a histogram with monotonically increasing values characterizes a rather compact shape). The structure histogram  $\mathbf{s}$  characterizes the structure of the shape by representing the evolution of the number of parts of the shape when going from the geodesic center towards larger eccentricity values (e.g. for a spider the number will range from 1 to 8, and for an ant from 1 and 6). Intuitively, one could say that  $\mathbf{h}$  looks at the widths of parts and  $\mathbf{s}$  at their number. Figure 8 shows the ECC based histograms for basic shapes, with and without holes and articulation.

*Effects of basic changes of the shape to the eccentricity histogram*<sup>5</sup>  $\mathbf{h}$ . The

---

<sup>5</sup>For the example class of shapes composed of 1D curves,  $\mathbf{h}(\mathcal{S})$  and  $\mathbf{s}(\mathcal{S})$  are equal (see



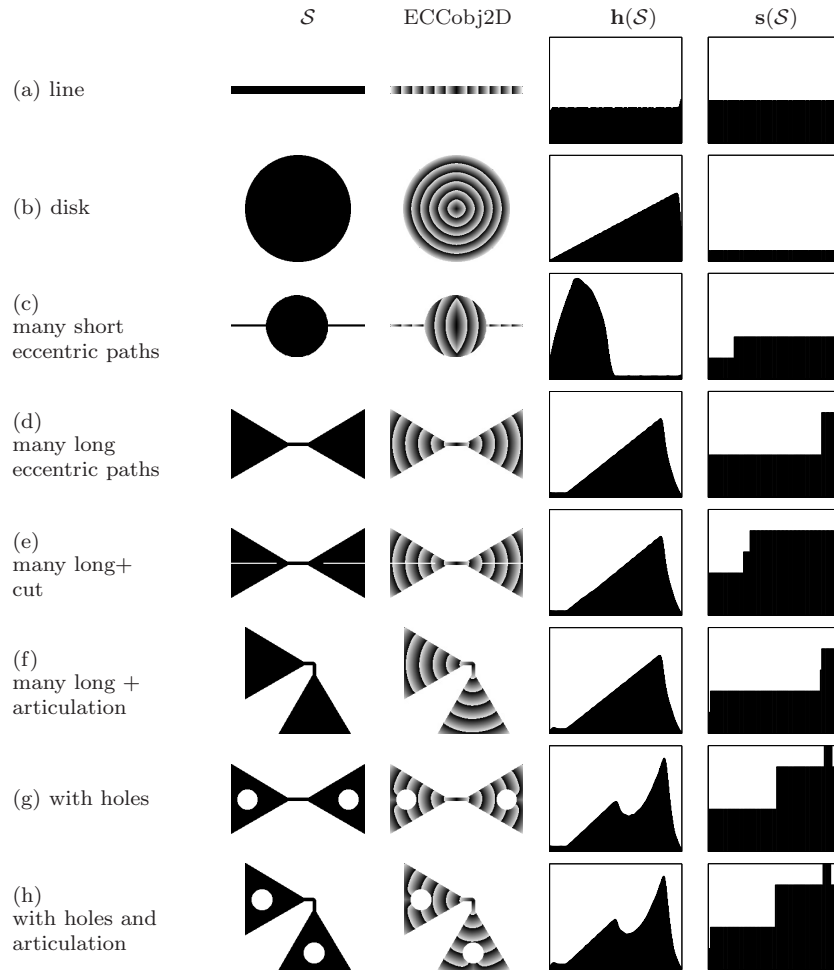


Figure 8: Basic shapes and their histograms  $h, s$ . (vertical axes independently scaled)

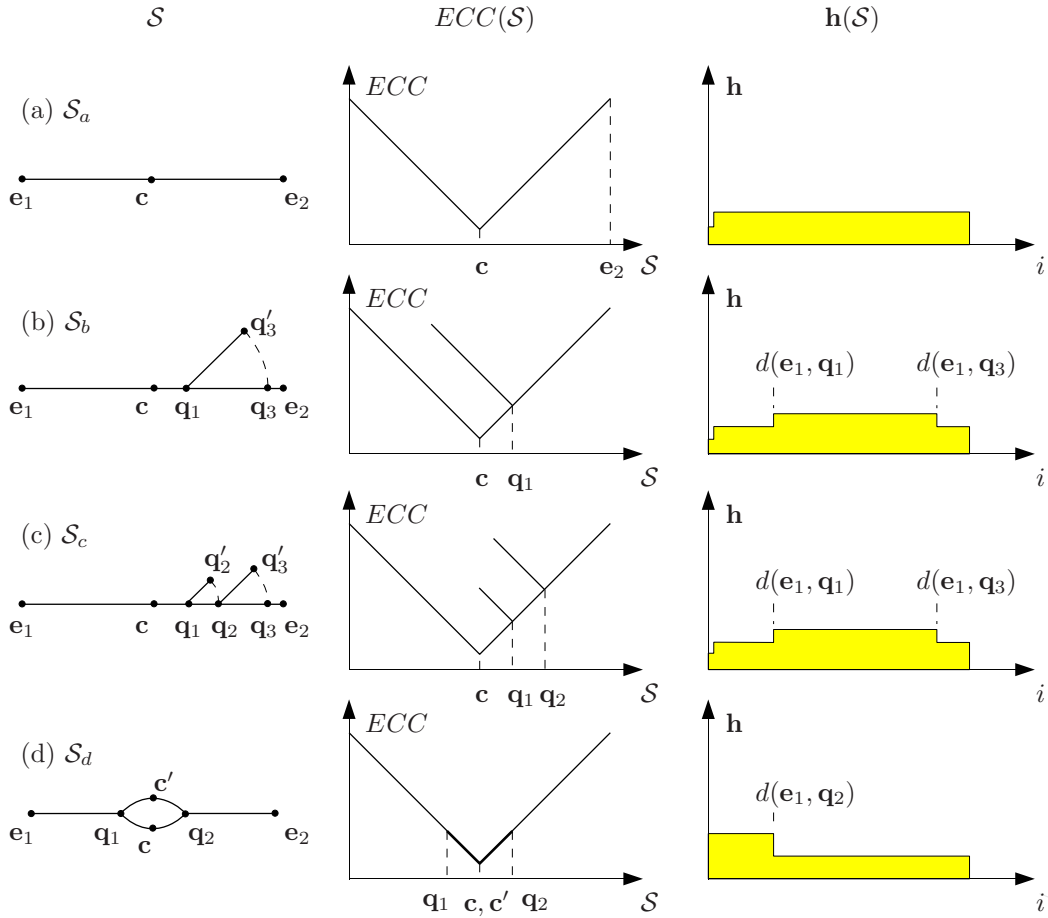


Figure 9: Behavior of eccentricity histograms for basic changes in the shape. Column  $\mathcal{S}$ : where possible, straight lines were used for illustration, but only the length of the curves is relevant, not whether they are straight or not. For the shown shapes (composed of curves)  $s(\mathcal{S}) = h(\mathcal{S})$  as each point of  $\mathcal{S}$  contributes with “1” to both of them: one eccentricity value and one connected component. (d) thick line in  $ECC(\mathcal{S})$ : between  $q_1$  and  $q_2$  two points have the same eccentricity.

histogram of the *ECC* of a simple open curve<sup>6</sup>  $\mathcal{S}_a$  with length  $l = d(\mathbf{e}_1, \mathbf{e}_2)$  (Figure 9(a)) is flat with a possibly smaller value in the first bin. The continuous formula is:

$$\mathbf{h}(\mathcal{S}_a, i) = \begin{cases} \frac{1}{l} & \text{if } i = \min(\text{ECC}(\mathcal{S}_a)) \\ \frac{2}{l} & \text{if } i > \min(\text{ECC}(\mathcal{S}_a)) \end{cases},$$

where  $\min(\text{ECC}(\mathcal{S}_a)) = d(\mathbf{e}_1, \mathbf{c}) = d(\mathbf{e}_2, \mathbf{c})$ .

Consider  $\mathcal{S}_b$  obtained by adding a simple open curve of length  $d(\mathbf{q}_1, \mathbf{q}'_3) < l/2$  connected at the point  $\mathbf{q}_1$  to  $\mathcal{S}_a$  (Figure 9(b)). Let  $\mathbf{q}_3 \in \mathcal{S}_b$  s.t.  $d(\mathbf{q}_1, \mathbf{q}_3) = d(\mathbf{q}_1, \mathbf{q}'_3)$  and  $d(\mathbf{q}_3, \mathbf{e}_1) = d(\mathbf{q}'_3, \mathbf{e}_1)$ . For the points with eccentricity between  $d(\mathbf{e}_1, \mathbf{q}_1)$  and  $d(\mathbf{e}_1, \mathbf{q}_3)$ , the eccentricity histogram of  $\mathcal{S}_b$  has increased by 50% (there is one additional point having each of the values in the domain). A shape without cycles (e.g.  $\mathcal{S}_a, \mathcal{S}_b, \mathcal{S}_c$ ) has only one center point (*ECC* minimum) and the histogram value for the center is always one. All other histogram values can be changed by adding branches as above.

The histogram value for the center can be changed by introducing cycles. Consider  $\mathcal{S}_d$  obtained by adding a simple open curve  $\mathbf{q}_1\mathbf{c}'\mathbf{q}_2$  of length  $\lambda(\mathbf{q}_1, \mathbf{q}_2)$  to  $\mathcal{S}_a$  (Figure 9(d)). The length  $d(\mathbf{e}_1, \mathbf{e}_2)$  is kept the same and  $\mathbf{q}_1\mathbf{q}_2$  has the same length if going over  $\mathbf{c}$  or  $\mathbf{c}'$ . Also  $d(\mathbf{e}_1, \mathbf{c}) = d(\mathbf{e}_1, \mathbf{c}') = d(\mathbf{e}_1, \mathbf{e}_2)/2$ . Two center points exist ( $\mathbf{c}$  and  $\mathbf{c}'$ ), and for the eccentricity values  $[d(\mathbf{c}, \mathbf{e}_1), d(\mathbf{q}_2, \mathbf{e}_1))$  there is one additional point.

For a given histogram, the steps used to create  $\mathcal{S}_b$  and  $\mathcal{S}_d$ , can be iterated to grow the continuous shape (for geodesics computed along thin lines). For discrete shapes, the number of points is finite<sup>7</sup>, which limits the number of curves that can be put close to each other and do not intersect. If the maximum shape size (number of pixels/voxels) and the number of bins  $k_h$  is fixed, not all (real valued) histograms can result as *ECC* histograms (it can also be seen as a discretization problem: the lower the resolution/maximum size, the higher the dependence between neighboring histogram bins).

*Equivalence classes of ECC based histograms.* A histogram has a smaller dimension (in our case 1D) than the shape, and a whole class of shapes is

---

Figure 9).

<sup>6</sup>The term *curve* is used to denote a one dimensional and continuous manifold, and includes both straight and non-straight lines.

<sup>7</sup>Depends on the discretization and maximum shape size.

projected into the same histogram. Two shapes  $\mathcal{S}$  and  $\tilde{\mathcal{S}}$  with the same eccentricity based histograms satisfy  $\Delta(\mathcal{S}, \tilde{\mathcal{S}}) = 0$ , and are thus considered to be the same according to our matching algorithm. Consider  $\mathcal{S}_c$  in Figure 9(c) obtained from  $\mathcal{S}_a$ , similar to  $\mathcal{S}_b$ , but with two curves s.t.  $d(\mathbf{q}_1, \mathbf{q}'_2) = d(\mathbf{q}_1, \mathbf{q}_2)$ ,  $d(\mathbf{q}_2, \mathbf{q}'_3) = d(\mathbf{q}_2, \mathbf{q}_3)$ , and  $d(\mathbf{q}_1, \mathbf{q}_3)$  is equal in both  $\mathcal{S}_b$  and  $\mathcal{S}_c$ . Hence, the two shapes  $\mathcal{S}_b$  and  $\mathcal{S}_c$  have the same eccentricity based histograms  $\mathbf{h}, \mathbf{s}$  and cannot be differentiated using only the histograms.

When the following operations are applied to a shape they create a shape in the same *equivalence class*, i.e. having the same or similar histograms:

- *scaling*;
- *rotation*;
- *isometric deformation* that does not change the structure of the shape (e.g. moving a finger, without touching another finger);
- *moving certain points*: simultaneous *thinning and thickening* of two parts with the same eccentricity values (e.g. making one finger thinner while making another one thicker);
- *moving certain parts*: taking a part and attaching it in another point with the same eccentricity while keeping the eccentricity values the same (e.g. moving a finger from the left hand to the right hand, obtaining a human with 4 respectively 6 fingers on his hands, also  $\mathcal{S}_b$  and  $\mathcal{S}_c$  in Figure 9);
- *certain changes in topology* without disconnecting the shape, when the contact points in the shape without the hole<sup>8</sup> have the same eccentricity value and no eccentric (longest) paths go over the created/destroyed connection (e.g. if considering the whole human as a shape, touching the tip of a finger with the neighboring finger, with the fingers as straight as possible). This change affects at least one bin of  $\mathbf{s}$ , but could be only minimally (one point contact as opposed to keeping two fingers touched along the whole length). Disconnected shapes cannot be handled by our descriptor.

---

<sup>8</sup>Points that will be connected to make the hole appear.

One can say that the eccentricity based histograms are influenced by the geometry and the structure of shapes, but they do not uniquely characterize them. This can be both a positive or a negative aspect, as one would like filter out for example noise, but probably not “attaching a second head”.

## 4. Matching Experiments in 2D and 3D

This section shows results on popular benchmarks and comparison with state of the art methods. When comparing the results, keep in mind that the proposed method is simple and matching the computed descriptors is fast. An approximation of the *ECC* can be computed for many shapes with as few as 50 distance propagations (e.g. the average number for the ECCmesh on the McGill database is 54), and determining  $\delta$  between two computed descriptors ( $L_2$ -norm) has practically no CPU time consumption<sup>9</sup>. Two fixed-length vectors as descriptors can be a very efficient indexing method. The approaches compared with are more complicated and include the decomposition of shapes, aligning or finding correspondences between features, etc.

### 4.1. 2D Shape Matching

For the experiments with 2D shapes we have used four shape databases: Kimia 25 [62] (Figure 13), Kimia 99 [52] (Figure 10), MPEG7 CE-Shape-1 [63] (Figure 11), and the articulated shape database of [8] (Ling articulated) (Figure 12). As a baseline we have added the results obtained by using the *Shape Index*, which is computed as the length of the boundary of the shape over its area.

A shape database is composed of  $q$  shapes  $\{\mathcal{S}_i\}_{i=1}^q$  and each shape  $\mathcal{S}_i$  has a label  $L(i) \in \{1, \dots, l_{\max}\}$ . Each label value  $1 \leq l \leq l_{\max}$  defines a class of shapes  $\mathcal{Q}(l) = \{\mathcal{S}_i \mid L(i) = l\}$ . The first columns of the blocks in Figure 13 show the shapes from the Kimia 25 database, ordered by classes (such as fish, planes, rabbits, etc.). Any shape matching algorithm  $\alpha$  assigns to each shape  $\mathcal{S}_i$  a vector of best matches  $\Phi_i$ , where  $\Phi_i(1)$  is the shape the most similar to  $\mathcal{S}_i$ ,  $\Phi_i(2)$  is the second hit, and so on. Depending on the benchmark,  $\Phi_i$  contains all shapes including the query shape  $\mathcal{S}_i$  (MPEG7), or leaves  $\mathcal{S}_i$  out,

---

<sup>9</sup>For example, on an Intel Xeon with 3 GHz, computing ECCmesh for the 255 shapes of the McGill database (see the Appendix) takes around 2.5 hours in total i.e. an average of 35 seconds per shape (code is partly matlab, partly c++). Computing  $\Delta$  for all pairs of descriptors (255\*254 matches) takes less than 1 second in total.

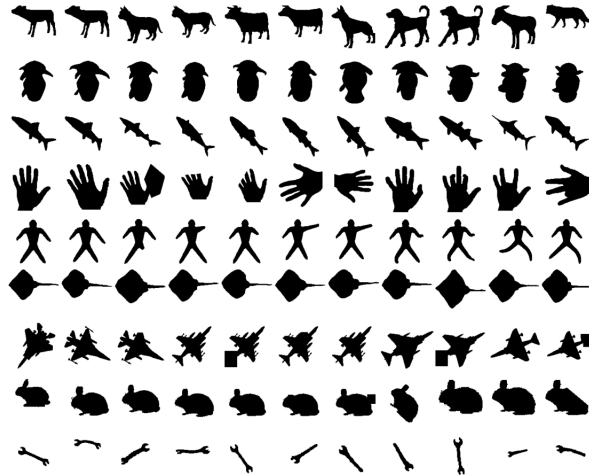


Figure 10: The 99 shapes from the Kimia 99 shape database. Shapes in the same row belong to the same class.

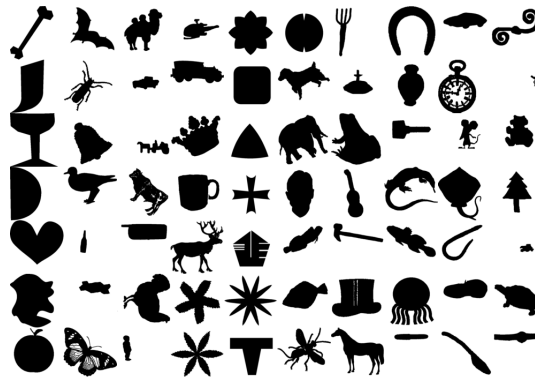


Figure 11: One shape from each of the 70 classes in the MPEG7 shape database.

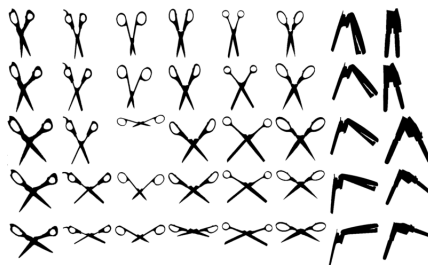


Figure 12: The 40 shapes from the Ling articulated database. Shapes in the same column belong to the same class.

i.e. the shape  $\mathcal{S}_i$  is not matched to itself and  $\Phi_i$  has  $q - 1$  elements (all other benchmarks presented).

For the Kimia 25 database  $l_{\max} = 6$  and  $q = 25$ , and for the Kimia 99 database,  $l_{\max} = 9$  and  $q = 99$ . The efficiency of various matching algorithms on Kimia databases is measured by the number of correct matches for each ranking position  $r$ :

$$\text{Match}_r(\Phi) \stackrel{\text{def.}}{=} \sum_{i=1}^q 1_{L(\Phi_i(r))=L(i)} \leq q. \quad (5)$$

Tables 2 and 3 give the value of  $\text{Match}_k$  for various shape matching algorithms. We also show results when using only one of the two: **h** or **s**.

In the case of the MPEG7 database, which contains  $l_{\max} = 70$  classes with 20 images each ( $q = 70 \times 20 = 1400$ ), the efficiency of matching algorithms is computed using the standard Bullseye test:

$$\begin{aligned} \text{Bullseye}(\Phi) &\stackrel{\text{def.}}{=} \frac{1}{20q} \sum_{r=1}^{40} \sum_{i=1}^q 1_{L(\Phi_i(r))=L(i)} \\ &= \frac{1}{20q} \sum_{r=1}^{40} \text{Match}_r(\Phi). \end{aligned} \quad (6)$$

This test counts the number of correct hits (same class) in the first 40 hits. For each image there can be at most 20 correct hits and a maximum of  $20 \times 1400$  hits can be obtained during the benchmark and thus  $\text{Bullseye}(\Phi) \leq 1$ . Table 4 gives the value of Bullseye for various shape matching algorithms.

The Ling articulated shape database was created specifically to show results in the presence of articulation. The database contains  $q = 40$  shapes of articulated objects from  $l_{\max} = 8$  classes. The efficiency of an algorithm is measured by the number of correct matches  $\text{Match}_r(\Phi)$  for each ranking position  $r$  (Equation 5). Table 5 gives the value of  $\text{Match}_k$  for various shape matching algorithms.

*Case study - Kimia 25.* Figure 13 shows the retrieval results for Kimia 25 when using only **h**, only **s** or both terms. The first columns show the 25 shapes  $\mathcal{S}_i$  (1-4 fish, 5-8 greebles, 9-13 hands, 14-17 airplanes, 18-21 rabbits, 22-25 tools). The following columns  $r$  show  $\Phi_i(r)$ , the rank- $r$  shape associated to  $\mathcal{S}_i$ .

Table 2: The value of  $\text{Match}_r(\Phi)$  for various algorithms on the Kimia 25 database ( $q = 25$  shapes from  $l_{\max} = 6$  classes).

Algorithm $\alpha$	$r=1$	2	3
Shape Index	14	10	10
<b>ECCobj2D, s only</b>	18	19	17
<b>ECCobj2D, h only</b>	20	16	14
<b>ECCobj2D</b>	22	20	17
Sharvit et al. [62]	23	21	20
Gdalyahu et al. [64]	25	21	19
Shape Context [7]	25	24	22
ID-Shape Context [8]	25	24	25

Table 3: The value of  $\text{Match}_r(\Phi)$  for various algorithms on the Kimia 99 database ( $q = 99$  shapes from  $l_{\max} = 9$  classes).

Algorithm $\alpha$	$r=1$	2	3	4	5	6	7	8	9	10
Shape Index	43	51	58	52	52	49	51	47	45	44
<b>ECCobj2D, s only</b>	84	68	65	67	56	57	51	50	41	31
<b>ECCobj2D, h only</b>	87	74	66	64	49	52	45	38	33	33
<b>ECCobj2D</b>	94	85	81	73	81	73	64	59	56	35
Shape Context [7]	97	91	88	85	84	77	75	66	56	37
Gen. Model [65]	99	97	99	98	96	96	94	83	75	48
Shock Edit [52]	99	99	99	98	98	97	96	95	93	82
ID-Shape Context [8]	99	99	99	98	98	97	97	98	94	79

Table 4: The value of  $\text{Bullseye}(\Phi)$  (Equation 6) for various algorithms on the MPEG7 database ( $q = 1400$  shapes from  $l_{\max} = 70$  classes).

Algorithm $\alpha$	$\text{Bullseye}(\Phi)$
random	2.86%
Shape Index	25.46%
<b>ECCobj2D</b>	54.56%
Shape Context [7]	64.59%
ID-Shape Context [8]	68.83%



Table 5: The value of  $\text{Match}_r(\Phi)$  for various algorithms on the Ling articulated shape database ( $q = 40$  shapes from  $l_{\max} = 8$  classes). See [8] for a description of these algorithms.

Algorithm $\alpha$	$r=1$	2	3	4
Shape Context+DP [8]	20	10	11	5
Shape Index	24	17	17	18
<b>ECCobj2D</b>	40	33	29	22
ID-Shape Context+DP [8]	40	34	35	27

The Kimia 25 database has shapes from 6 classes: 5 classes with 4 images each, and one (hands) with 5 images (1 simulating a segmentation error). If considering only  $\mathbf{h}$  (Figure 13 top) the class with the best results are rabbits, followed by tools, hands, fishes, airplanes and greebles. Two questions immediately rise when looking at these results:

1. Why are the greebles considered to be more similar to the hands than to other greebles?
2. Why does a rabbit appear in so many cases when the matching has failed?

For the first question, consider the ECC histograms  $\mathbf{h}$  of the greebles and the not occluded hands (Figure 14). The histograms are similar even though the shapes are of different classes, e.g. the histogram of the first greeble (Figure 14 top-left) looks more similar to the hands, than the second and third greeble. This is due to the abstraction of a 2D shape to a 1D histogram which, in the case of  $\mathbf{h}$ , disregards certain structural properties of distances/paths (studied in detail in Section 3.4). The structure histogram  $\mathbf{s}$  can better discriminate between these shapes, and adding it to the descriptor compensates for this effect.

For the second question, consider the shapes in Figure 15 (a rabbit -  $\mathcal{S}_{19}$ , and two tools -  $\mathcal{S}_{25}$  and  $\mathcal{S}_{22}$ ). When matching  $\mathcal{S}_{25}$ , the rabbit has a smaller distance of  $\mathbf{h}$  than  $\mathcal{S}_{22}$ , even though one might say that the histograms  $\mathbf{h}$  of  $\mathcal{S}_{25}$  and  $\mathcal{S}_{22}$  reveal more similar distance characteristics than the histogram  $\mathbf{h}$  of  $\mathcal{S}_{19}$  (see Figure 15). Both  $\mathcal{S}_{25}$  and  $\mathcal{S}_{22}$  have more long distances than medium, and short, while  $\mathcal{S}_{19}$  has a peak in the medium. This effect is due to *typical histogram matching methods*, which are inherently low level and *fail to capture the high level context of the task*. Note that similar examples could be constructed also using  $\mathbf{s}$ . Discussion follows in Section 5.

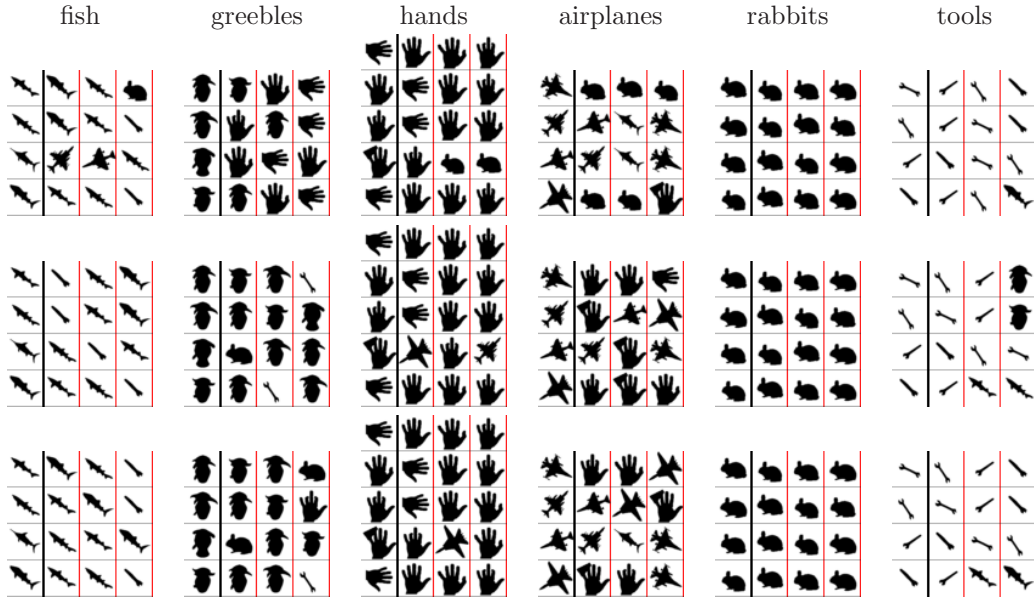


Figure 13: Retrieval results on Kimia 25 using top: **h**, middle: **s**, bottom: both **h** and **s**. Each row shows the query shape, followed by the first three matches.

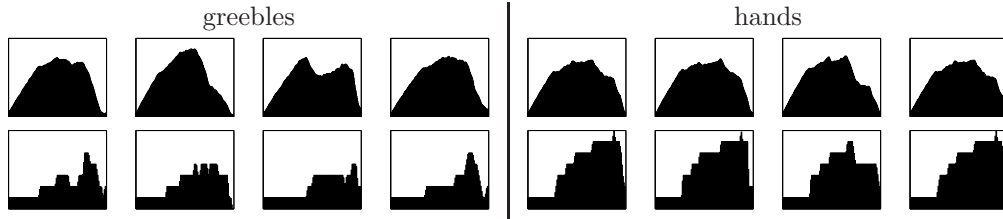


Figure 14: Histograms for greebles and not occluded hands, top: **h**, bottom **s**.

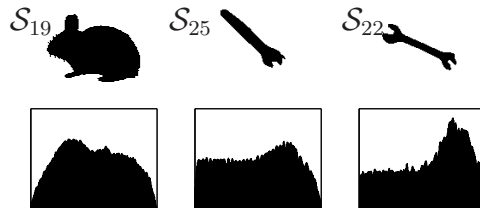


Figure 15: Three shapes from the Kimia 25 database and their eccentricity histograms **h**.

Global geometrical and structural statistics of the shapes are well captured by our low dimensional descriptors. However, more detailed structural properties like part decomposition, ordering and adjacency of parts, and the geometrical features corresponding to these parts are not considered by our signature extraction.

#### 4.2. 3D Articulated Shape Matching

A widely used 3D object retrieval database is the Princeton Shape Benchmark [66]. It contains 1,814 3D object models organized by class and is effective for comparing the performance of a variety of methods. However, the majority of the models corresponds to rigid, man-made objects. Only a limited number of shapes in the database have articulated parts. As one of the main advantages of using the eccentricity transform is its robustness with respect to articulation, we have turned to the McGill Shape Benchmark [53]. It contains several models from the Princeton repository and others added by the authors. The main advantage of this benchmark is that from its 455 3D shapes, 255 have significant part articulation. We show the results on the  $q = 255$  shapes grouped into the  $l_{\max} = 10$  classes of articulated shapes (Figure 16). Shapes are not matched to themselves and so  $\Phi_i$  contains  $q - 1$  shapes.

We compare results with:

- medial surfaces (MS) [46];
- spherical harmonic descriptor (HS) [67];
- shape distributions (SD) [16].

MS computes the medial surfaces of a voxelized shape and decomposes them in parts. The similarity of two shapes is obtained from the matching of the directed acyclic graphs describing their parts. HS transforms rotation dependent spherical shape descriptors into rotation invariant ones. It describes spherical functions on the shape in terms of the amount of energy they contain at different frequencies. The  $L_2$ -norm is used to match two descriptors. SD computes the similarity of two shapes by the comparison of two probability distributions sampled from a shape function measuring geometric properties of the 3D model. Best results are obtained with a function called D2 which represents the distribution of Euclidean distances between pairs of randomly selected points on the surface of a 3D model.

Three *ECC* based descriptors are evaluated (Figure 4):

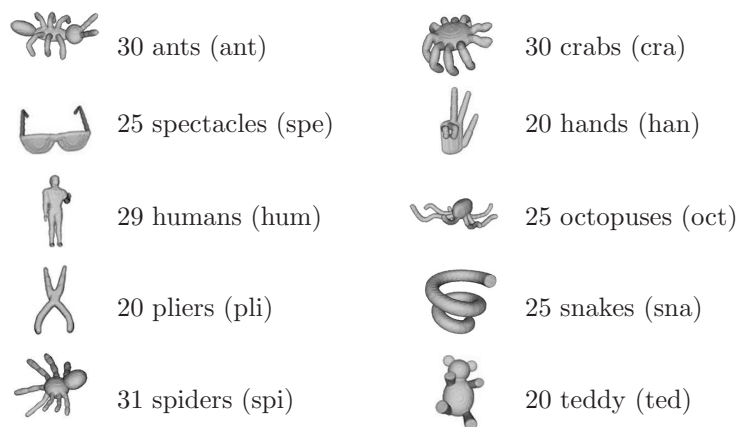


Figure 16: The object classes from the McGill 3D shape database having significant part articulation. The number of instances in each class are given.

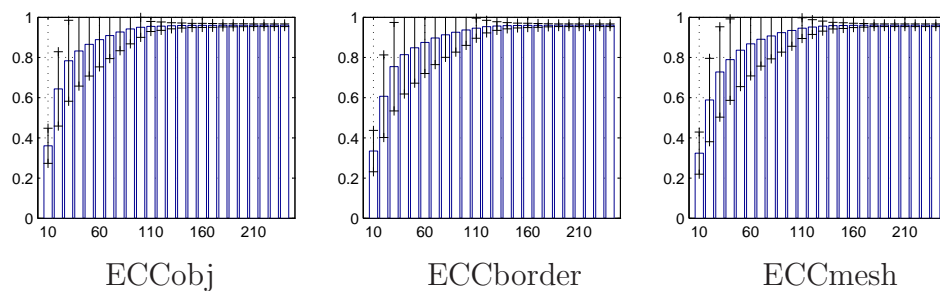


Figure 17: Recall for several rank thresholds

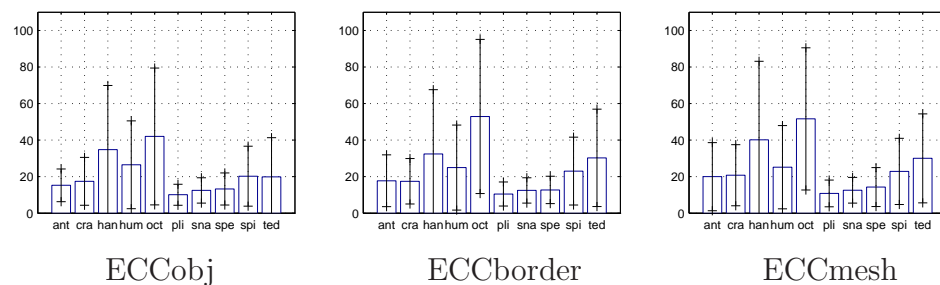


Figure 18: Average ranks for each class. The first three letters of each class name are printed.

1. **ECCobj** - eccentricity of the whole shape (all object voxels);
2. **ECCborder** - eccentricity of the border/boundary voxels;
3. **ECCmesh** - eccentricity of the triangular mesh of the surface of the shape.

ECCborder is the eccentricity transform  $ECC(\partial_6\mathcal{S})$ , where  $\partial_6\mathcal{S}$  is the 6 connected voxel boundary of  $\mathcal{S}$ . ECCmesh is computed on the 2D manifold defined on the boundary of the 3D shapes. ECCborder uses distance computation in the 3D volume, ECCmesh in the 2D surface. If the resolution of the shapes is increased, ECCborder and ECCmesh converge to the same value. For a similar resolution, ECCmesh needs less memory, as cells not part of the boundary do not have to be stored (e.g. interior of the shape), and it can be more accurate when approximating the eccentricity of the surface, as the computation is done on the surface itself, not on an approximating (thicker) volume.

The size of the voxel models used for ECCobj and ECCborder is below  $128 \times 128 \times 128$  (the size of the binary 3D images is  $128 \times 128 \times 128$ ).

*Experimental results.* In the following, results of the three variants on shapes of the 10 articulated classes of the McGill Shape Benchmark are given. The notation from Section 4.1 is used. The following measures are considered.

$$\text{Recall}(\Phi_i, t) = \frac{1}{|\mathcal{Q}(L(i))| - 1} \sum_{r=1}^t 1_{L(\Phi_i(r))=L(i)}$$

The recall computes the ratio of models in the database in the same category as the query, with indexing rank  $\leq t$ , to the total number of shapes in the same category (never including the query itself). The average results and standard deviation for several rank thresholds ( $t = 10, 20, \dots$ ), over all classes, are given in Figure 17.

$$\text{AvgRank}(\Phi_i) = \frac{1}{|\mathcal{Q}(L(i))| - 1} \sum_{r=1}^{q-1} r \cdot 1_{L(\Phi_i(r))=L(i)}$$

For all queries in a class, the average of the ranks of all other shapes in that class are computed. Figure 18 shows the average and the standard deviation of the ranks for each class (lower average is better).

Table 6 shows the average score for all pairs of classes. Each shape in the database is matched against all other shapes and each cell shows the

average of the score (Equation 4) between all combinations of shapes of the two classes defined by the row and column.

$$\text{Precision}(\Phi_i, t) = \frac{1}{t} \sum_{r=1}^t 1_{L(\Phi_i(r))=L(i)}$$

Precision refers to the ratio of the relevant shapes retrieved, to the total number retrieved. Figure 19 shows the *precision-recall* curves for each of the 10 classes. Precision-recall curves are produced by varying the parameter  $t$ . Better results are characterized by curves closer to the top, i.e. recall = 1 for all values of precision. Precision and recall are common in information retrieval for evaluating retrieval performance. They are usually used where static document sets can be assumed. However, they are also used in dynamic environments such as web page retrieval [68].

As can be seen in Figures 17, 18, and 19, and Table 6, ECCobj does in most cases a better job than ECCborder and ECCmesh. The recall of the three methods is very similar, with slightly better results from ECCobj (noticeable on the first three bars). With respect to the average ranks, ECCobj does better with the ants, octopus, spiders, teddy, is equal to one of ECCborder and ECCmesh with the crabs, humans, pliers, snakes, and is slightly worse than one of the other methods with the humans and spectacles. None of the three variants produces an average class rank higher than 21% of 255 (average rank 42, 53, and 52, for the octopus, for ECCobj, ECCborder, and ECCmesh, respectively). ECCobj has the smallest average class distance (highest similarity) correct for all 10 classes. For both ECCborder and ECCmesh the smallest average class distance is correct for 9 classes, while the correct class is the second smallest one for the remaining class, the octopus (see Table 6). A discussion considering the differences between the three ECC variants follows at the end of this section.

Figure 19 shows comparative precision-recall results of ECCobj, ECCborder, ECCmesh, MS, HS, and SD. ECCobj, ECCborder, and ECCmesh are close, except for the teddy bears, where ECCobj is superior to the other two. The best results (higher precision vs. recall) are reached by the *ECC* variants for the snakes, by MS for the crabs, and HS and SD for teddy. For these best results the MS and the ECCobj have almost identical precision-recall, followed by HS and SD (Figure 20). The worst results are achieved by ECCobj, ECCborder, and ECCmesh for the octopus, MS for the pliers, HS for snakes, and SD for the hands. The results of MS for the pliers are superior

Table 6: Average matching results multiplied by 100 (smaller means more similar). For each row, the first and second smallest value are printed in bold.

		ants	crabs	hands	humans	octopus	pliers	snakes	spectacles	spiders	teddy
ECCobj	ants	<b>2.99</b>	7.74	7.93	8.15	8.31	7.90	15.73	19.49	<b>7.25</b>	10.57
	crabs	7.74	<b>2.60</b>	9.09	9.85	5.88	10.37	20.85	20.17	<b>4.75</b>	11.24
	hands	7.93	9.09	<b>4.39</b>	<b>5.34</b>	10.15	5.36	10.83	13.02	9.76	5.72
	humans	8.15	9.85	5.34	<b>3.62</b>	10.11	<b>4.08</b>	8.88	10.60	9.51	5.21
	octopus	8.31	5.88	10.15	10.11	<b>5.10</b>	10.46	19.94	17.55	<b>5.60</b>	11.97
	pliers	7.90	10.37	5.36	<b>4.08</b>	10.46	<b>1.45</b>	6.58	9.44	10.44	4.53
	snakes	15.73	20.85	10.83	8.88	19.94	6.58	<b>1.40</b>	<b>6.29</b>	20.78	7.80
	spectacles	19.49	20.17	13.02	10.60	17.55	9.44	<b>6.29</b>	<b>3.00</b>	19.60	9.29
	spiders	7.25	<b>4.75</b>	9.76	9.51	5.60	10.44	20.78	19.60	<b>2.98</b>	11.91
	teddy	10.57	11.24	5.72	5.21	11.97	<b>4.53</b>	7.80	9.29	11.91	<b>3.47</b>
ECCborder	ants	<b>2.40</b>	5.62	6.51	6.39	<b>4.83</b>	6.26	14.33	16.84	4.93	8.32
	crabs	5.62	<b>2.58</b>	7.83	9.27	5.78	10.14	19.65	22.26	<b>4.71</b>	11.20
	hands	6.51	7.83	<b>3.97</b>	<b>4.91</b>	7.97	5.01	10.09	12.33	8.30	5.60
	humans	6.39	9.27	4.91	<b>3.01</b>	8.73	<b>3.62</b>	7.73	10.37	8.68	4.53
	octopus	4.83	5.78	7.97	8.73	<b>4.49</b>	8.48	17.74	18.12	<b>4.44</b>	10.57
	pliers	6.26	10.14	5.01	<b>3.62</b>	8.48	<b>1.57</b>	6.45	8.84	8.85	4.08
	snakes	14.33	19.65	10.09	7.73	17.74	6.45	<b>1.49</b>	<b>5.63</b>	18.89	6.35
	spectacles	16.84	22.26	12.33	10.37	18.12	8.84	<b>5.63</b>	<b>2.60</b>	19.70	8.92
	spiders	4.93	4.71	8.30	8.68	<b>4.44</b>	8.85	18.89	19.70	<b>2.80</b>	10.72
	teddy	8.32	11.20	5.60	4.53	10.57	<b>4.08</b>	6.35	8.92	10.72	<b>3.74</b>
ECCmesh	ants	<b>2.45</b>	5.90	5.86	6.31	<b>4.52</b>	6.01	13.76	15.62	4.77	7.31
	crabs	5.90	<b>2.86</b>	8.02	9.62	5.52	10.42	20.03	22.16	<b>4.49</b>	10.77
	hands	5.86	8.02	<b>4.20</b>	5.19	7.39	<b>4.97</b>	10.38	11.98	7.74	5.50
	humans	6.31	9.62	5.19	<b>3.01</b>	8.67	<b>3.61</b>	7.41	9.78	8.74	4.53
	octopus	4.52	5.52	7.39	8.67	<b>4.19</b>	8.39	17.63	18.20	<b>4.16</b>	9.61
	pliers	6.01	10.42	4.97	<b>3.61</b>	8.39	<b>1.71</b>	6.43	8.30	8.76	3.87
	snakes	13.76	20.03	10.38	7.41	17.63	6.43	<b>1.56</b>	<b>4.73</b>	18.76	6.87
	spectacles	15.62	22.16	11.98	9.78	18.20	8.30	<b>4.73</b>	<b>2.58</b>	19.69	8.78
	spiders	4.77	4.49	7.74	8.74	<b>4.16</b>	8.76	18.76	19.69	<b>2.65</b>	9.75
	teddy	7.31	10.77	5.50	4.53	9.61	<b>3.87</b>	6.87	8.78	9.75	<b>3.70</b>

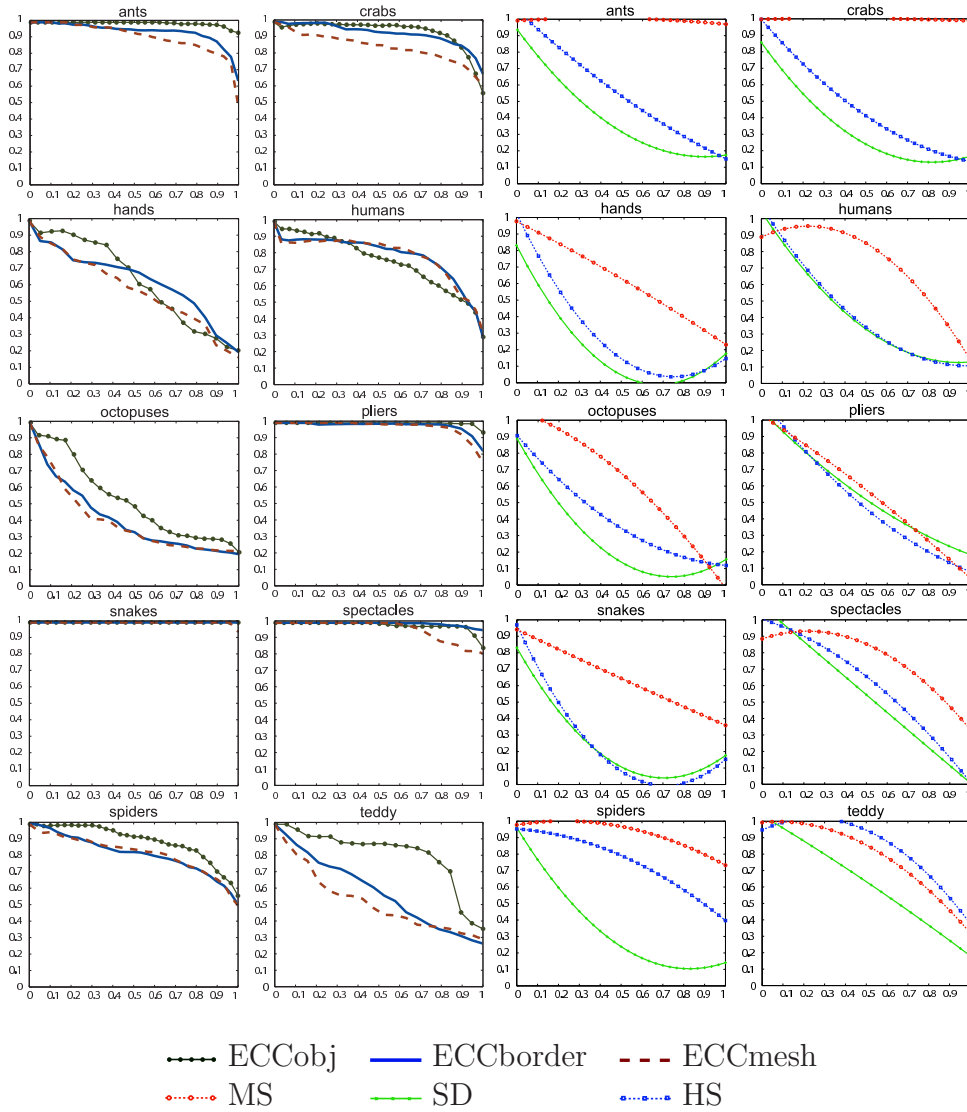


Figure 19: Precision-recall for the articulated shapes from the McGill dataset ( $q = 255$  shapes from  $l_{\max} = 10$  classes). Left two columns: ECCobj, ECCborder, ECCmesh. Right two columns (image taken from [46], with kind permission of Springer Science and Business Media): results of three other methods on the same database: medial surfaces (MS) [46], spherical harmonic descriptor (HS) [67], and shape distributions (SD) [16]. Precision: horizontal axis, recall: vertical axis. (Best visualized in color.)



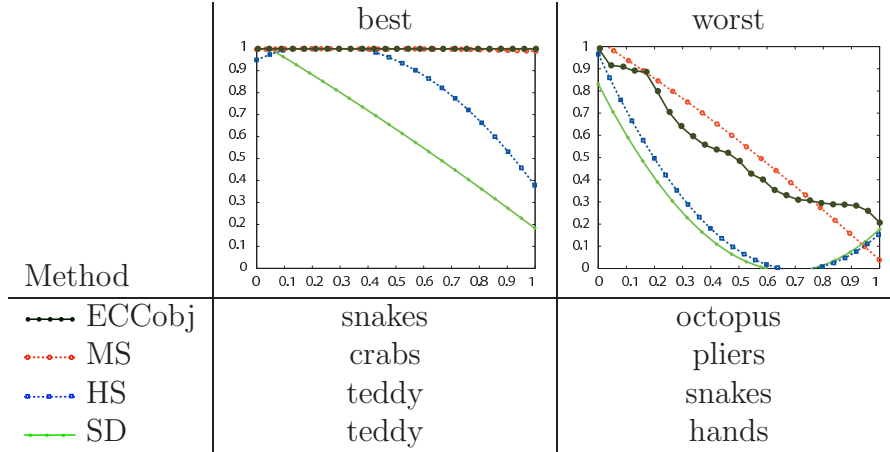


Figure 20: Best and worst precision-recall curves for the results of ECCObj, MS, HS, and SD (Picture synthesized from Figure 19). (Best visualized in color.)

to ECCObj for the octopus, which are in turn superior to the HS, superior to SD (Figure 20). In comparison to all other three methods (MS, HS, SD), the eccentricity based methods score better on the pliers, spectacles and snakes, and ECCObj is on par with the best on ants, hands, and spiders.

MS searches for a concrete mapping between the parts of articulated shapes, which explains the overall superior results. The *ECC* variants are articulation insensitive and able to characterize the compactness (geometry) of the shapes quite well. However, describing the structure of the shape using a histogram does not capture the connections between the parts (e.g. one could detach a finger from a hand and attached it to the other one, without changing the descriptor of the human). The descriptor computed by HS is invariant to the independent rotation of parts, even if the shape gets disconnected. SD uses a function which is based on Euclidean distance and not invariant to articulation.

The differences in the results of ECCObj vs. ECCborder and ECCmesh, can be linked to the compactness of the shapes and the width of their joints. During articulation, the variation of the geodesic distances is larger when computed on the ‘skin’ (boundary) compared to computed inside the shape (smaller). In the case of 2D shapes the eccentricity of the boundary is a constant. In 3D it manages to capture some of the properties of the shape, but it is more unstable, e.g. the eccentricity transform of a simply connected

volume has in most cases a single stable center (minimum), while the eccentricity transform of its boundary will have a disconnected center or at least one with a more complex structure. A more concrete example of such instability with only minor deformation of the shape is the ECC of the surface of:

- a sphere: all points make up the center;
- an ellipsoid with two equal radiuses and a slightly longer third one: the center is one circle.

As a last experiment we show results of ECCmesh on the Robustness Benchmark from the SHREC'10 dataset [69]<sup>10</sup>. The dataset consists of 1184 shapes given as triangle meshes. The query set consists of 13 shape classes taken from a subset of the dataset with simulated transformations applied to them. For each shape, transformations are split into 10 classes (isometry, topology, small and big holes, global and local scaling, noise, shot noise, partial occlusion, sampling, and a combination of all transformations). In each class, the transformation appears in five different strength levels. The total number of transformations per shape is 55 plus the null shape (neutral pose), and the total query set size is 728 shapes. The reported results are mean average precision (mAP).

The given triangle meshes are possibly disconnected and non-manifold (e.g. more than 2 triangles can share an edge). As the *ECC* is computed on a single connected component and our geodesic distance computation requires manifolds, for each shape we select the largest subset of triangles (based on area), which are connected and approximate a manifold.

Table 7 gives the results of ECCmesh on the SHREC'10 robustness benchmark. The transformation “partial” reduces the shape to a disconnected subset of the original triangles, consisting of many components. Our method cannot handle such disconnected shapes and computes the descriptor of one of the connected mesh parts instead. Thus we also give the average score over only the transforms which keep the query shape connected (without “partial” or “all”). When computing the shape descriptors we did not refine the meshes or sample from them uniformly, but took the *ECC* values for the given vertices of the mesh. The resolution of the meshes is not uniform and

---

<sup>10</sup>See [http://tosca.cs.technion.ac.il/book/shrec\\_robustness.html](http://tosca.cs.technion.ac.il/book/shrec_robustness.html) for images of the shapes in the dataset.

Table 7: Retrieval results on the SHREC’10 Robustness Benchmark (total  $q = 1184$  shapes, with a query set of 728 shapes from  $l_{\max} = 13$  classes). The reported performance is mean average precision. Rows indicate the type of transform that the query shape was altered with, columns indicate the maximum strength of the applied transform. “avg”: results over all queries in the dataset. “avg (connected)”: results only with connected shapes (without transformations “partial” and “all”).

transformation	strength				
	1	$\leq 2$	$\leq 3$	$\leq 4$	$\leq 5$
isometry	94.23	88.62	86.66	81.82	81.10
topology	80.45	67.73	65.26	64.57	64.84
holes	85.26	80.45	81.28	78.14	73.40
microholes	94.23	94.23	94.23	94.23	94.23
scale	87.01	90.62	91.82	90.18	89.20
localscale	94.23	91.67	88.61	84.92	79.47
sampling	88.46	77.12	58.15	43.97	35.26
noise	94.23	94.23	92.95	91.99	90.64
shotnoise	94.23	94.23	94.23	92.31	89.74
partial	9.61	8.20	7.53	6.53	6.17
all	0.63	13.34	36.25	41.00	37.61
avg	94.11	89.68	86.84	83.50	80.33
avg (connected)	98.06	95.52	93.02	90.13	87.54

explains why the results for “scale” are not 100%, as expected from the normalization. The sensitivity of the histograms to the non-uniform sampling of the vertices is reflected in the results for the transformation “sampling”<sup>11</sup>. For an evaluation of different methods on the SHREC’10 benchmark see [69].

#### 4.3. Discussion: 2D and 3D

The computed shape similarities are *robust with respect to scaling, rotation, and part articulation*. The matching results are good, especially when considering the *straightforward approach*, and databases of shapes undergoing articulation. Our histogram representation is invariant to articulation, thus achieving state of the art performance on these databases. In contrast, the most efficient shape matching algorithms [8, 46] are more complicated and require extraction of salient features and local signatures that need to be

---

<sup>11</sup>Refining the meshes to obtain a uniform sampling for the locations of the vertices could in principle be added as a pre-processing step.

aligned or registered. E.g. the method in [8] describes shapes by a collection of local descriptors sampled at different locations on one selected boundary of the shape. Each descriptor is a 2-dimensional histogram, and is based on the computation of geodesic distances and the angle made by the tangent to the geodesic with the tangent to the shape boundary at the point of the descriptor. Similarities are computed using dynamic programming as the cost of the best matching of the descriptors along the boundary. In [46], first a medial surface is obtained by doing topology preserving thinning on the average outward flux computed on the distance transform of the shape. The obtained medial surface is segmented into components and using saliency a directed acyclic graph (DAG) is built to describe the shape. To match a query DAG with a database, first an indexing strategy is used to quickly select a few candidates, followed by a bipartite graph matching to compute node correspondences and the matching score. In addition to the matching score both methods [8, 46] can provide correspondences between the used parts/points. Also the higher quality results of [8] can be explained by the usage of local descriptors together with a score computation which considers their order – this gives more detailed geometric information with a higher tolerance to missing parts.

The *limitations* of our approach include: (1) Eccentricity structure histograms  $\mathbf{s}$  are not able to *fully capture the part structure* of the shape and thus histograms of different shapes can be very similar. (2) *Histogram ‘matching’* (whether using the  $L_2$ -norm or more sophisticated methods) is inherently low level and *does not consider the higher level context* in which it is applied.

One can identify the limitations discussed before (see Figure 19 and Table 6): for classes with simple topology (e.g. snakes and spectacles), the results are very good. For classes where part decomposition and detailed geometry play an important role (e.g. octopus, hands, humans), the discrimination capabilities are reduced.

#### 4.4. The Parameters $k_h, k_s$ and $w$

The approach has three parameters: the number of bins  $k_h, k_s$ , of the histograms  $\mathbf{h}, \mathbf{s}$ , and the mixing factor  $w$ . In all presented experiments 2D and 3D,  $\mathbf{k}_h = \mathbf{200}$  and  $\mathbf{k}_s = \mathbf{100}$  were used. The values were chosen based on a few initial trials on a small set of shapes (Kimia25). Table 8 shows the results on Kimia99 for different values of the parameter  $k_h$  when using only  $\mathbf{h}$  as a shape descriptor, i.e. when  $w = 1$ . For  $\mathbf{s}$ , a difference of at least 1 (2D, and voxel representations in 3D) respectively  $\sqrt{2}$  is required between

Table 8: The value of  $\text{Match}_r(\Phi)$  for **ECCobj2D**, **h** only ( $w = 1$  in Equation 4), on the Kimia 99 database using descriptors with different number of bins. See Table 3 for comparison with other methods.

number of bins $k$	$r=1$	2	3	4	5	6	7	8	9	10
10	85	69	66	59	52	51	40	46	42	30
25	87	74	68	63	48	53	45	38	35	33
50	87	74	67	68	45	51	43	38	37	31
100	87	74	66	65	48	53	44	38	34	32
200	87	74	66	65	49	51	45	39	33	33
500	87	74	67	64	48	53	45	38	33	33
1000	87	74	67	64	48	54	44	37	34	33
2000	87	74	67	64	48	53	44	38	34	33

Table 9: The value of  $\text{Match}_r(\Phi)$ ,  $r = 1, 2, 3$  for **ECCobj2D** on the Kimia 25, Kimia 99, and Ling databases using  $k_h = 200$ ,  $k_s = 100$  and different values of  $w$ . See Section 4.1 for comparison with other methods.

database	$w$	$r=1$	2	3
Kimia 25	0.93	22	19	16
Kimia 99	0.93	95	84	79
Ling	0.93	40	33	27
Kimia 25	0.94	22	20	17
Kimia 99	0.94	94	85	81
Ling	0.94	40	33	29
Kimia 25	0.95	22	20	18
Kimia 99	0.95	93	86	81
Ling	0.95	40	33	29

the centers of two neighboring bins. Thus any value of  $k_s$  larger than half the average of the geodesic diameters of the shapes is enough. The used value for  $w$  was 0.94 and was selected based on a few trials on the Kimia25, Kimia99, and Ling databases (note that  $\mathbf{h}$  and  $\mathbf{s}$  use a different normalization scheme, which explains the apparently large bias toward  $\mathbf{h}$ ). Table 9 shows results for different values of  $w$ .

As the shapes are discrete, the number of distance values of the *ECC* is finite. Let  $\mathbf{h}^c$  be the ordered set of eccentricity values computed for a shape  $\mathcal{S}$ , i.e. each distinct value that exists in the *ECC* of the discrete shape  $\mathcal{S}$ . We have  $\min(\mathbf{h}^c)$  equal to the *ECC* value of the center (minimum *ECC*) and greater or equal to half the diameter of the shape ( $\max(\mathbf{h}^c) = \max(\text{ECC})$ ). The largest distance between two neighboring (grid) points is equal to one (shapes are required to be 4 respectively 6 connected). For the *ECC* histogram of a shape not to contain any empty bins, the number of bins  $k_h$  has to satisfy:

$$k \leq \max(\text{ECC}(\mathcal{S})) - \min(\text{ECC}(\mathcal{S})).$$

Depending on the shape,  $k_h$  could be much higher and still have no empty bins in  $\mathbf{h}$ , e.g. for  $\mathcal{S}$  a disk with radius  $u$  in  $\mathbb{Z}^2$  and the Euclidean distance, there are more distinct values than  $u$  (consider the discrete approximation of the Euclidean circle). An absolute upper bound is  $k_h = |\mathcal{S}|$ . If this number is exceeded, there will be empty bins in  $\mathbf{h}$ .

As  $k_h$  decreases, the description capability of the histogram also decreases. In the extreme case, **a single bin** would just contain  $|\mathcal{S}|$ , and for the normalized histogram it would contain the value 1. **Two bins** can give the equivalent of a simple compactness measure (similar to the *circularity ratio* or *shape index*), which relates the area of the shape to the area of the circle with the same diameter). **Three bins** could be considered as a relative measure for short/long/medium distances and can characterize more than the simple compactness measure.

A higher number of bins increases the dimension of the space in which distances are computed, and gives more flexibility in the relations, e.g. in 2D there can be maximum 3 points s.t. they are pairwise at the same Euclidean distance (equilateral triangle), and this number increases to 4 in 3D (regular tetrahedron). Assuming that the number of classes and their pairwise (average) distances are known, a lower bound for the number of bins is equal to the smallest dimension in which the classes can be embedded s.t. the

distances computed in the histogram space are equal or close to the given distances.

## 5. Potential Extensions

*Topological changes in the shape.* Geodesic distances are sensitive to changes in the topology of the shape. For example, it is enough to touch the index fingers of two hands in one point to drastically change the geodesic distance between the points of the two palms: before we had to go over the arms and torso, now we only have to travel over two fingers. This problem has been approached in [42, 41] by using diffusion distance as an alternative to geodesic distance. Diffusion distances consider all possible paths by which two points can be connected and are expected to be more stable to changes in topology (in the previous example, the high number of paths going through the thicker arms will still be dominant over the path(s) going over the small size contact area between the two fingers). The definition of the eccentricity transform (Equation 2) is defined over a metric space and diffusion distances could also be used.

*Describing and matching part structure.* One of the problems identified in Section 3.4 and during the experiments (Sections 4.1 and 4.2) is that the histograms  $\mathbf{s}$  do not capture the exact structure/topology of the shape. Classical methods to describe the topology of a shape (e.g. Reeb graphs [70], and homology generators [71]) do not capture geometrical aspects. An approach to deal with this problem is presented in [72]. Two descriptors are used to describe a shape: a geometric one, based on the *Global Geodesic Function* (GGF) [34], which is defined for a point as the sum of the geodesic distances to all points of the shape multiplied by a factor, and a topological one, the Reeb graph of the shape using the GGF as the Morse function.

Initial steps in combining the eccentricity transform with Reeb graphs have been presented in [73].

*A better histogram matching.* The problem of having a matching function that is aware of the context in which it is applied can be approached in two ways: (1) use expert knowledge about the context to create an algorithm that considers the proper features, or (2) learn the important features by giving a set of representative examples (e.g. [45]). In [74, 75], a survey of current distance metric learning methods is given. The purpose of distance

metric learning is to learn a distance metric for a space, from a given collection of pairs of similar/dissimilar points. The learned distance is supposed to preserve the distance relation among the training data. Example training data would be:  $\mathcal{S}_1$  is more similar to  $\mathcal{S}_2$  than to  $\mathcal{S}_3$ . The result is a distance function that would replace the  $L_2$ -norm in Equation 3 with a new measure, which is adapted to the task of computing the distance of eccentricity histograms as given by the training examples.

*Partial shape similarity.* The presented approach matches “whole” shapes and will work in the presence of only minor occlusion. Even though the eccentricity values remain the same when certain parts of a shape are removed, removing parts will change the histograms, proportional to the size of the removed parts. Partial shape similarity could be approached by first decomposing a shape into parts (like in the approaches in [12, 46]) and then matching the parts to each other. This way one can obtain part correspondences and higher robustness with respect to occlusion.

*Higher dimensional data.* 4D data has started to be available in the medical image processing community (e.g. 3D scans of a beating heart, over time). The presented method is general and should work in any metric space. This includes 4D, but also gray scale images (e.g. gray values can determine the distance propagation speed in the respective cells). A study in this direction is planned.

## 6. Conclusion

We have presented a method for matching 2D and 3D shapes. The method is based on the eccentricity transform, which uses maximal geodesic distances and is insensitive to articulation. Descriptors are composed of two terms: a normalized histogram of the eccentricity transform and a histogram of the connected components of the level-sets of the eccentricity transform. They characterize the compactness and structure of the shape, are compact and easy to match. The method is straightforward but still efficient, achieving state of the art results on databases of shapes undergoing articulation. Articulation is indeed the main invariance provided by our histogram representation. Experimental results on popular 2D and 3D shape matching benchmarks are given, with computation on binary 2D images, binary 3D voxel shapes, and 3D triangular meshes. The parameters of the method, and the relation between changes to a shape and its corresponding eccentricity



histogram are discussed in detail. Adding extended structural information to the descriptor and searching for correspondences between parts would help to overcome the cases where shapes of different classes have similar histograms. Matching can be improved by adding context in the histogram similarity computation e.g. through metric distance learning.

## 7. Acknowledgments

The presented work was partially supported by the Austrian Science Fund under grants S9103-N13 and P18716-N13. Adrian Ion was supported, in part, by the European Commission, under project MCEXT-025481.

## Appendix

For completeness, the algorithm *ECC06* [47] used to compute the eccentricity transform for the shapes in our experiments is included (see [48] for an analysis of the speed/error performance). *ECC06* (see Algorithm 1) tries to identify points of the geodesic center (minimum *ECC*) and use those to find eccentric point candidates. Computing  $D^{\mathcal{S}}(\mathbf{c})$  for a center point  $\mathbf{c} \in C(\mathcal{S})$  is expected to create local maxima where eccentric points lie. In a first phase, the algorithm identifies at least two diameter ends by repeatedly ‘jumping’ (computing  $D^{\mathcal{S}}(\mathbf{p})$ ) for the point that had the highest value in the previous estimation. In the second phase, the center points  $\mathbf{c}_i$  are estimated as the points with the minimum eccentricity and all local maxima  $\mathbf{m}$  of  $D^{\mathcal{S}}(\mathbf{c})$  are marked as eccentric point candidates. For all  $\mathbf{m}$ ,  $D^{\mathcal{S}}(\mathbf{m})$  is computed and accumulated. When no new local maxima are found (i.e. with  $D^{\mathcal{S}}(\mathbf{m})$  not previously computed), the algorithm stops.

## References

- [1] B. Bustos, D. A. Keim, D. Saupe, T. Schreck, D. V. Vranić, Feature-based similarity search in 3D object databases, *ACM Comput. Surv.* 37 (4) (2005) 345–387.
- [2] R. C. Veltkamp, L. Latecki, Properties and performance of shape similarity measures, in: *Proceedings of the 10th IFCS Conference on Data Science and Classification*, Slovenia, 2006.

---

**Algorithm 1**  $ECC06(\mathcal{S})$  - Estimate eccentricity transform by progressive refinement.

---

**Input:** Discrete shape  $\mathcal{S}$ .

```

1: for all  $\mathbf{q} \in \mathcal{S}$ ,  $ECC(\mathbf{q}) \leftarrow 0$  /*initialize distance matrix*/
2:  $\mathbf{p} \leftarrow$  random point of  $\mathcal{S}$  /*find a starting point*/
3:
4: /*Phase 1: find a diameter*/
5: while  $\mathbf{p}$  not computed do
6:    $ECC \leftarrow \max\{ECC, D^{\mathcal{S}}(\mathbf{p})\}$  /*accumulate & mark  $\mathbf{p}$  as computed*/
7:    $\mathbf{p} \leftarrow \arg \max\{ECC(\mathbf{p}) \mid \mathbf{p} \in \mathcal{S}\}$  /*highest current ECC (farthest away)*/
8: end while
9:
10: /*Phase 2: find center points and local maxima*/
11:  $pECC \leftarrow 0$  /*make sure we enter the loop*/
12: while  $pECC \neq ECC$  do
13:    $pECC \leftarrow ECC$ 
14:    $C \leftarrow \arg \min\{ECC(\mathbf{p}) \mid \mathbf{p} \in \mathcal{S}\}$  /*points with min. ECC*/
15:   for all  $\mathbf{c} \in C$ ,  $\mathbf{c}$  not computed do
16:      $D \leftarrow D^{\mathcal{S}}(\mathbf{c})$  /*compute distances from the center*/
17:      $ECC \leftarrow \max\{ECC, D\}$  /*accumulate & mark  $\mathbf{c}$  as computed*/
18:
19:      $\mathcal{M} \leftarrow \{\mathbf{q} \in \mathcal{S} \mid D(\mathbf{q}) \text{ local max. in } \mathcal{S} \ \& \ \mathbf{q} \text{ not computed}\}$ 
20:     for all  $\mathbf{m} \in \mathcal{M}$ ,  $\mathbf{m}$  not computed do
21:        $ECC \leftarrow \max\{ECC, D^{\mathcal{S}}(\mathbf{m})\}$  /*accumulate & mark  $\mathbf{m}$  as computed*/
22:     end for
23:   end for
24: end while

```

**Output:** Distances  $ECC$ .

---

- [3] C. T. Zahn, R. Z. Roskies, Fourier descriptors for plane closed curves, *IEEE Transactions on Computer* 21 (3) (1972) 269–281.
- [4] A. M. Bronstein, M. M. Bronstein, A. M. Bruckstein, R. Kimmel, Analysis of two-dimensional non-rigid shapes, *International Journal of Computer Vision* 78 (1) (2008) 67–88.
- [5] A. B. Hamza, H. Krim, Geodesic matching of triangulated surfaces, *IEEE Transactions on Image Processing* 15 (8) (2006) 2249–2258.
- [6] F. Mokhtarian, A. K. Mackworth, A theory of multiscale, curvature-based shape representation for planar curves, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14 (8) (1992) 789–805.
- [7] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (4) (2002) 509–522.
- [8] H. Ling, D. W. Jacobs, Shape classification using the inner-distance, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (2) (2007) 286–299.
- [9] K. Nasreddine, A. Benzinou, R. Fablet, Shape geodesics for boundary-based object recognition and retrieval, 2009, pp. 405–408.
- [10] J. Xu, Shape matching using morphological structural shape components, 2008, pp. 2596–2599.
- [11] M. Reuter, F.-E. Wolter, N. Peinecke, Laplace-spectra as fingerprints for shape matching, in: L. Kobbelt, V. Shapiro (Eds.), *Symposium on Solid and Physical Modeling*, ACM, 2005, pp. 101–106.
- [12] K. Siddiqi, A. Shokoufandeh, S. Dickinson, S. W. Zucker, Shock graphs and shape matching., *International Journal of Computer Vision* 30 (1999) 1–24.
- [13] C. Xu, J. Liu, X. Tang, 2d shape matching by contour flexibility, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31 (1) (2009) 180–186.

- [14] Y. W. Chen, C. L. Xu, Rolling penetrate descriptor for shape-based image retrieval and object recognition, *Pattern Recognition Letters* 30 (9) (2009) 799 – 804, advanced Intelligent Computing Theory and Methodology, International Conference on Intelligent Computing.
- [15] L. Gorelick, M. Galun, E. Sharon, R. Basri, A. Brandt, Shape representation and classification using the poisson equation, in: *CVPR* (2), 2004, pp. 61–67.
- [16] R. Osada, T. Funkhouser, B. Chazelle, D. Dobkin, Shape distributions, *ACM Transactions on Graphics* 21 (4) (2002) 807–832.
- [17] M. Elad, A. Tal, S. Ar, Content based retrieval of vrml objects: an iterative and interactive approach, in: *Eurographics Workshop on Multimedia*, Springer, New York, 2001, pp. 107–118.
- [18] E. Paquet, A. Murching, T. Naveen, A. Tabatabai, M. Rioux, Description of shape information for 2-D and 3-D objects, *Signal Processing: Image Communication* 16 (1–2) (2000) 103–122.
- [19] F. A. Sadjadi, E. L. Hall, Three-dimensional moment invariants, *Pattern Analysis and Machine Intelligence*, *IEEE Transactions on PAMI-2* (2) (1980) 127 –136.
- [20] C. Y. Ip, L. Regli, W. C. Sieger, A. Shokoufandeh, Automated learning of model classifications, in: *8th ACM Symposium on Solid Modeling and Applications*, ACM Press, New York, 2003, pp. 322–327.
- [21] K. Guo, M. Li, 3D gray level moment invariants: A novel shape representation, 2009, pp. 1 –4.
- [22] D. Vranic, D. Saupe, Description of 3D-shape using a complex function on the sphere, in: *International Conference on Multimedia and Expo*, IEEE, 2002, pp. 177–180.
- [23] D. Vranic, D. Saupe, 3D model retrieval with spherical harmonics and moments., in: *23rd DAGM-Symposium on Pattern Recognition*, Springer, London, UK, 2001, pp. 392–397.
- [24] M. Ankerst, G. Kastenmüller, H.-P. Kriegel, T. Seidl, 3D shape histograms for similarity search and classification in spatial databases,

- in: 6th International Symposium on Advances in Spatial Databases, Springer, London, UK, 1999, pp. 201–226.
- [25] D. Vranic, D. Saupe, 3D shape descriptor based on 3D fourier transform, in: EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services, Comenius University, 2001, pp. 271–274.
  - [26] J. W. H. Tangelder, R. C. Veltkamp, Polyhedral model retrieval using weighted point sets, in: Shape Modeling International, IEEE, Seoul, Korea, 2003, pp. 119–129.
  - [27] A. Mademlis, P. Daras, D. Tzovaras, M. G. Strintzis, 3D object retrieval using the 3D shape impact descriptor, *Pattern Recognition* 42 (11) (2009) 2447 – 2459.
  - [28] M. R. Ruggeri, G. Patané, M. Spagnuolo, D. Saupe, Spectral-driven isometry-invariant matching of 3D shapes, *International Journal of Computer Vision* 2009. doi:10.1007/s11263-009-0250-0.
  - [29] E. Paquet, M. Rioux, Nefertiti: A tool for 3-D shape databases management, *SAE transactions* 108 (1999) 387–393.
  - [30] T. Zaharia, F. Preux, Three-dimensional shape-based retrieval within the mpeg-7 framework, in: SPIE Conference on Nonlinear Image Processing and Pattern Analysis XII, 2001, pp. 133–145.
  - [31] T. Ansary, J.-P. Vandeborre, S. Mahmoudi, M. Daoudi, A bayesian framework for 3D models retrieval based on characteristic views, 2nd International Symposium on 3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004 (2004) 139–146.
  - [32] C. M. Cyr, B. B. Kimia, A similarity-based aspect-graph approach to 3D object recognition, *International Journal of Computer Vision* 57 (1) (2004) 5–22.
  - [33] D.-Y. Chen, M. Ouhyoung, X.-P. Tian, Y.-T. Shen, M. Ouhyoung, On visual similarity based 3D model retrieval, in: Eurographics, Granada, Spain, 2003, pp. 223–232.
  - [34] M. Hilaga, Y. Shinagawa, T. Kohmura, T. L. Kunii, Topology matching for fully automatic similarity estimation of 3D shapes, in: SIGGRAPH

- '01: Proceedings of the 28th Conference on Computer Graphics and Interactive Techniques, ACM, New York, USA, 2001, pp. 203–212.
- [35] Y. Shinagawa, T. Kunii, Y. Kergosien, Surface coding based on morse theory, *Computer Graphics and Applications*, IEEE 11 (5) (1991) 66–78.
  - [36] H. Sundar, D. Silver, N. Gagvani, S. Dickinson, Skeleton based shape matching and retrieval, *Shape Modeling International*, 2003 (2003) 130–139.
  - [37] A. Johnson, M. Hebert, Using spin images for efficient object recognition in cluttered 3D scenes, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21 (5) (May 1999) 433–449.
  - [38] M. Reuter, Hierarchical shape segmentation and registration via topological features of laplace-beltrami eigenfunctions, *International Journal of Computer Vision* 2009. doi:10.1007/s11263-009-0278-1.
  - [39] F. Mémoli, G. Sapiro, A theoretical and computational framework for isometry invariant recognition of point cloud data, *Foundations of Computational Mathematics* 5 (3) (2005) 313–347.
  - [40] F. Mémoli, Spectral gromov-wasserstein distances for shape matching, in: *Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment (ICCV workshop, NORDIA'09)*, 2009.
  - [41] A. M. Bronstein, M. M. Bronstein, M. Mahmoudi, R. Kimmel, G. Sapiro, A gromov-hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching, *International Journal of Computer Vision* 89/2-3.
  - [42] R. M. Rustamov, Laplace-beltrami eigenfunctions for deformation invariant shape representation, in: A. G. Belyaev, M. Garland (Eds.), *Symposium on Geometry Processing*, Vol. 257 of ACM International Conference Proceeding Series, Eurographics Association, 2007, pp. 225–233.
  - [43] A. Ion, G. Peyré, Y. Haxhimusa, S. Peltier, W. G. Kropatsch, L. Cohen, Shape matching using the geodesic eccentricity transform - a study, in: C. B. W. Ponweiser, M. Vincze (Ed.), *The 31st Annual Workshop of the*

Austrian Association for Pattern Recognition (OAGM/AAPR), OCG, Schloss Krumbach, Austria, 2007, pp. 97–104.

- [44] A. Ion, N. M. Artner, G. Peyré, S. B. López Mármol, W. G. Kropatsch, L. Cohen, 3D shape matching by geodesic eccentricity, in: Workshop on Search in 3D (in conjunction with CVPR 2008), IEEE, Anchorage, Alaska, 2008.
- [45] A. M. Bronstein, M. M. Bronstein, L. J. Guibas, M. Ovsjanikov, Shape google: geometric words and expressions for invariant shape retrieval, ACM Transactions on Graphics (to appear).
- [46] K. Siddiqi, J. Zhang, D. Macrini, A. Shokoufandeh, S. Bouix, S. J. Dickinson, Retrieving articulated 3-D models using medial surfaces, Machine Vision and Applications 19 (4) (2008) 261–275.
- [47] W. G. Kropatsch, A. Ion, Y. Haxhimusa, T. Flanitzer, The eccentricity transform (of a digital shape), in: 13th International Conference on Discrete Geometry for Computer Imagery (DGCI), Springer, Szeged, Hungary, 2006, pp. 437–448.
- [48] A. Ion, W. G. Kropatsch, E. Andres, Euclidean eccentricity transform by discrete arc paving, in: D. Coeurjolly, I. Sivignon, L. Tougne, F. Dupont (Eds.), 14th IAPR International Conference on Discrete Geometry for Computer Imagery (DGCI), Vol. LNCS 4992 of Lecture Notes in Computer Science, Springer, Lyon, France, 2008, pp. 213–224.
- [49] S. Manay, D. Cremers, B.-W. Hong, A. J. Yezzi, S. Soatto, Integral invariants for shape matching, IEEE Trans. Pattern Anal. Mach. Intell. 28 (10) (2006) 1602–1618.
- [50] P. Soille, Morphological Image Analysis, 2nd Edition, Springer, 2002.
- [51] A. Ion, The eccentricity transform of  $n$ -dimensional shapes with and without boundary, Ph.D. thesis, Vienna Univ. of Technology, Faculty of Informatics (2009).
- [52] T. B. Sebastian, P. N. Klein, B. B. Kimia, Recognition of shapes by editing their shock graphs., IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (5) (2004) 550–571.

- [53] J. Zhang, K. Siddiqi, D. Macrini, A. Shokoufandeh, S. J. Dickinson, Retrieving articulated 3-D models using medial surfaces and their graph spectra, in: 5th International Workshop Energy Minimization Methods in Computer Vision and Pattern Recognition, EMMCVPR 2005, Vol. 3757 of LNCS, Springer, 2005, pp. 285–300.
- [54] S. Suri, The all-geodesic-furthest neighbor problem for simple polygons., in: Symposium on Computational Geometry, 1987, pp. 64–75.
- [55] F. Maisonneuve, M. Schmitt, An efficient algorithm to compute the hexagonal and dodecagonal propagation function., *Acta Stereologica* 8 (2) (1989) 515–520.
- [56] M. Schmitt, Propagation function: Towards constant time algorithms, *Acta Stereologica: Proceedings of the 6th European Congress for Stereology*, September 7-10, Prague 13 (2).
- [57] F. Harary, *Graph Theory*, Addison Wesley, 1969.
- [58] R. Diestel, *Graph Theory.*, Springer, New York, 1997.
- [59] R. Klette, A. Rosenfeld, *Digital Geometry*, Morgan Kaufmann, 2004.
- [60] J. Sethian, *Level Sets Methods and Fast Marching Methods*, 2nd Edition, Cambridge Univ. Press, 1999.
- [61] E. Andres, M.-A. Jacob, The discrete analytical hyperspheres, *IEEE Transactions on Visualization and Computer Graphics* 3 (1) (1997) 75–86.
- [62] D. Sharvit, J. Chan, H. Tek, B. Kimia, Symmetry-based indexing of image databases, in: *IEEE Workshop on Content-based Access of Image and Video Libraries*, 1998, pp. 56–62.
- [63] L. J. Latecki, R. Lakämper, U. Eckhardt, Shape descriptors for non-rigid shapes with a single closed contour., in: *IEEE International Conference on Computer Vision and Pattern Recognition*, Hilton Head, SC, USA, 2000, pp. 1424–1429.
- [64] Y. Gdalyahu, D. Weinshall, Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes, *IEEE Trans. Pattern Anal. Mach. Intell.* 21 (12) (1999) 1312–1328.



- [65] Z. Tu, A. L. Yuille, Shape matching and recognition - using generative models and informative features, in: *Computer Vision - ECCV 2004*, 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part III, Vol. 3023 of Lecture Notes in Computer Science, Springer, 2004, pp. 195–209.
- [66] P. Shilane, P. Min, M. M. Kazhdan, T. A. Funkhouser, The princeton shape benchmark, in: *International Conference on Shape Modeling and Applications (SMI)*, Genova, Italy, IEEE Computer Society, 2004, pp. 167–178.
- [67] M. Kazhdan, T. Funkhouser, S. Rusinkiewicz, Rotation invariant spherical harmonic representation of 3D shape descriptors, in: *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, Eurographics Association, 2003, pp. 156–164.
- [68] T. Fawcett, An introduction to ROC analysis, *Pattern Recognition Letters* 27 (8) (2006) 861–874.
- [69] A. M. Bronstein, M. M. Bronstein, U. Castellani, B. Falcidieno, A. Fusiello, A. Godil, L. J. Guibas, I. Kokkinos, Z. Lian, M. Ovsjanikov, G. Patane, M. Spagnuolo, R. Toldo, Shrec 2010: robust large-scale shape retrieval benchmark, in: *EUROGRAPHICS Workshop on 3D Object Retrieval (3DOR)*, 2010.
- [70] G. Reeb, Sur les points singuliers d'une forme de pfaff complément intégrable ou d'une fonction numérique, *Annales de l'institut Fourier*, 14 no. 1 14 (1) (1964) 37–42.
- [71] J. R. Munkres, *Elements of Algebraic Topology*, Addison-Wesley, 1993.
- [72] D. Aouada, D. W. Dreisigmeyer, H. Krim, Geometric modeling of rigid and non-rigid 3D shapes using the global geodesic function, in: *NORDIA workshop in conjunction with IEEE International Conference on Computer Vision and Pattern Recognition (CVPR08)*, IEEE, Anchorage, Alaska, USA, 2008.
- [73] A. Ion, S. Peltier, S. Alayrangues, W. G. Kropatsch, Eccentricity based topological feature extraction, in: S. Alayrangues, G. Damiand, L. Fuchts, P. Lienhardt (Eds.), *Workshop on Computational Topology in Image Context*, Poitiers, France, 2008.

- [74] L. Yang, R. Jin, Distance metric learning: A comprehensive survey, Tech. rep., Department of Computer Science and Engineering, Michigan State University, [http://www.cse.msu.edu/~yangliu1/frame\\_survey\\_v2.pdf](http://www.cse.msu.edu/~yangliu1/frame_survey_v2.pdf) (2006).
- [75] L. Yang, R. Jin, R. Sukthankar, Y. Liu, An efficient algorithm for local distance metric learning, in: The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA, AAAI Press, 2006.