Matching 2D and 3D Articulated Shapes using Eccentricity

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

International Journal of Computer Vision manuscript No.
(will be inserted by the editor)

Matching 2D & 3D Articulated Shapes using Eccentricity

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

Received: date / Accepted: date

Abstract Shape matching should be invariant to the typical intra-class deformations present in nature. The majority of shape descriptors are quite complex and not invariant to the deformation or articulation of object parts. Geodesic distances computed over a 2D or 3D shape are articulation insensitive. The eccentricity transform considers the length of the longest geodesics. It is robust with respect to Salt and Pepper noise, and minor segmentation errors, and is stable in the presence of holes. We present a method for 2D and 3D shape matching based on the eccentricity transform. Eccentricity histograms make up descriptors insensitive to rotation, scaling, and articulation. The descriptor is highly compact and the method is straight-forward. Experimental results on established 2D and 3D benchmarks show results comparable to more complex state of the art methods. Properties and results are discussed in detail.

Partially supported by the Austrian Science Fund under grants S9103-N13 and P18716-N13.

Adrian Ion
PRIP, Vienna University of Technology
E-mail: ion@prip.tuwien.ac.at

Nicole M. Artner
ARC, Smart Systems Division, Austria
E-mail: nicole.artner@arcs.ac.at

Gabriel Peyré
CNRS, CEREMADE, Université Paris-Dauphine
E-mail: peyre@ceremade.dauphine.fr

Walter G. Kropatsch
PRIP, Vienna University of Technology
E-mail: krw@prip.tuwien.ac.at

Laurent Cohen
CNRS, CEREMADE, Université Paris-Dauphine
E-mail: cohen@ceremade.dauphine.fr

Keywords Eccentricity Transform · Shape Matching · Articulation · Geodesic Distance

1 Introduction

The recent increase in available 3D models and acquisition systems has seen 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 (Bustos et al, 2005).

Shape matching requires to set up a signature that characterizes the properties of interest for the recognition (Veltkamp and Latecki, 2006). The invariance of this signature to local deformations such as articulations is important for the identification of 2D and 3D shapes. Matching can then be carried out over this reduced space of signatures. Most shape descriptors are computed over a transformed domain that amplifies the important features of the object while throwing away ambiguities such as translation, rotation or local deformations.

For 2D shapes, the *Fourier transform of the boundary curve* (Zahn and Roskies, 1972) is an example of such a transformed domain descriptor adapted to smooth shapes. Shape *transformations computed with geodesic distances* (Bronstein et al, 2006) lead to signatures invariant to isometric deformations such as bending or articulation. To capture salient features of 2D objects, local quantities such as *curvature* (Mokhtarian and Mackworth, 1992) or *shape contexts* (Belongie

et al, 2002) can be computed. They can be extended to *bending invariant signatures* using geodesic distances (Ling and Jacobs, 2007). More global features include the *Laplace spectra* (Reuter et al, 2005) and the *skeleton* (Siddiqi et al, 1999). Some transformations involve the computation of a function defined on the shape, for instance the solution to a linear *partial differential equation* (Gorelick et al, 2004) or *geometric quantities* (Osada et al, 2002). Geodesic distance information such as the *mean-geodesic transform* (Hamza and Krim, 2003) could also be used.

Among appraoches matching 3D shapes, existing approaches can be divided into (Bustos et al, 2005): *Statistical* descriptors, like for example geometric *3D moments* employed by Elad et al (2001); Paquet et al (2000), and the *shape distribution* (Osada et al, 2002; Ip et al, 2003). *Extension-based* descriptors, which are calculated from features sampled along certain directions from a position within the object (Vranic and Saupe, 2002, 2001a). *Volume-based* descriptors use the volumetric representation of a 3D object to extract features (examples are *Shape histograms*, Ankerst et al (1999), *Model Voxelization*, Vranic and Saupe (2001b), and point set methods, Tangelder and Veltkamp (2003)). Descriptors using the *surface geometry* compute curvature measures and/or the distribution of surface normal vectors (Paquet and Rioux, 1999; Zaharia and Preux, 2001). *Image-based* descriptors reduce the problem of 3D shape matching to an image similarity problem by comparing 2D projections of the 3D objects (Ansary et al, 2004; Cyr and Kimia, 2004; Chen et al, 2003). Methods matching the *topology* of two objects (for example *Reeb* graphs, where the topology of the 3D object is described by a graph structure, Hilaga et al (2001); Shinagawa et al (1991)). *Skeletons* are intuitive object descriptions and can be obtained from a 3D object by applying a thinning algorithm on the voxelization of a solid object like in Sundar et al (2003). Descriptors using *spin images* work with a set of 2D histograms of the object geometry and a search for point-to-point correspondences is done to match 3D objects (Johnson and Hebert, May 1999).

The majority of shape descriptors is quite complex and not invariant to the deformation or articulation of object parts.

In Ling and Jacobs (2007) 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 it 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 relative change of the geodesic distance[1] during articulation is small and that geodesic distances are articulation insensitive.

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 in addition robust with respect to Salt and Pepper noise and minor segmentation errors (Kropatsch et al, 2006), and stable in the presence of holes. Normalized histograms of the eccentricity transform are invariant to changes in orientation, scale, and articulation. We propose eccentricity histograms as descriptors for 2D and 3D shape matching. They require only a simple representation and can be efficiently matched. Initial results have been presented for 2D in Ion et al (2007), and for 3D (volumetric representation) in Ion et al (2008a). This article presents a common framework with an in dept analysis of the properties of the approach, supported by promising experimental results and detailed discussion. Four variants of the descriptor are used, one for 2D shapes and three for 3D shapes (volume, border voxels, mesh) and compared to state of the art methods. To the best of our knowledge, this is the first approach applying eccentricity (furthest point distance) to the problem of shape matching. The presented approach could be fitted to either of the categories *extension-based* or *volume-based*, and it is a *transformation computed with geodesic distances*.

Like the method in (Ling and Jacobs, 2007), our method does not involve any part models. The articulation model is only for the analysis of the properties of the geodesic distance. 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) which relies on the correct decomposition of the unknown object into parts. A one-to-one mapping is not always possible as some parts might be missing due to, for example, segmentation errors. Decomposition of the shapes into parts is not required by our approach.

The paper is organized as follows: Section 2 recalls the eccentricity transform and discusses used variants and computation. Section 3 gives the proposed matching method and discusses pros and cons of the descriptor (Section 3.1). Section 4 presents details and dis-

[1] Called 'inner-distance' in Ling and Jacobs (2007).

cusses the results of the experiments, followed by parameters and improvements in Section 5. Section 6 concludes the paper.

2 Eccentricity Transform

The following definitions and properties follow Kropatsch et al (2006); Ion et al (2008b), and are extended to n-dimensional domains.

Let the shape $\mathcal{S}$ be a closed set in $\mathbb{R}^n$. A path π 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 $\pi \in \Pi(\mathbf{p}_1,\mathbf{p}_2)$

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

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

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

and $\pi(t)$ is a parametrization of the path from $\mathbf{p}_1 = \pi(0)$ to $\mathbf{p}_2 = \pi(1)$. Any shortest path $\nu \in \Pi(\mathbf{p}_1,\mathbf{p}_2)$, $\lambda(\nu) = d(\mathbf{p}_1,\mathbf{p}_2)$ is called a geodesic (path).

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

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

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 (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 definition of length. For a definition of the ECC of a graph see Kropatsch et al (2006).

The ECC is quasi-invariant to articulated motion and robust against salt and pepper noise (which creates holes in the shape) (Kropatsch et al, 2006). An analysis of the variation of geodesic distance under articulation can be found in Ling and Jacobs (2007).

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 $\mathcal{S}$ (Kropatsch et al, 2006). The *center* is the set of points that have the smallest (minimum) eccentricity. The *diameter* of a shape $\mathcal{S}$ is the maximum ECC, which is the length of the longest geodesic path.

This paper considers the class 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 triangle meshes representing the surface of the 6-connected 3D shapes. 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). If increasing the resolution of the shapes, ECCborder and ECCmesh converge to the same value.

2.1 Computation

One of the first attempts to deal with the problem of furthest point is presented in Suri (1987), where an algorithm for finding the eccentric/furthest vertices for the vertices of a simple polygon is given in. Later Maisonneuve and Schmitt (1989); Schmitt (1993) proposed an efficient algorithm for simply connected shapes on the hexagonal and dodecagonal grid. The concept of eccentricity of a vertex can be found in classical graph theory books (Harary, 1969; Diestel, 1997), and the concept of eccentricity transform[2] in recent discrete geometry (Klette and Rosenfeld, 2004) and mathematical morphology books (Soille, 2002). Computation is not discussed and no references to holes in a shape are made.

The straight forward computation approach is: for each point of $\mathcal{S}$, compute the distance to all other points and take the maximum. In Ion et al (2008b) faster computation and efficient approximation algorithms are presented. For this paper, the fastest one, algorithm ECC06 (see Appendix), has been used.

ECC06 relies on the computation of the *shape bounded single source distance transform*[3] $D^{\mathcal{S}}(\mathbf{p})$ (Figure 1(b)), which is computed for estimated eccentric point candidates in an iterative manner. $D^{\mathcal{S}}(\mathbf{p})$ associates to each point $\mathbf{q} \in \mathcal{S}$ the geodesic distance to $\mathbf{p}$. $D^{\mathcal{S}}$ can be computed using Fast Marching (Sethian, 1999), which allows for an efficient computation in $O(N * \log(N))$ steps, for $N = |\mathcal{S}|$ grid points. The complexity for computing $ECC(\mathcal{S})$ using ECC06 and Fast Marching is $O(K * N * \log(N))$ where $K \leqslant |\partial\mathcal{S}|$ is adapted to the shape.

[2] Known in the mathematical morphology community as the *propagation function*.

[3] Also called geodesic distance function with marker set $\mathbf{p}$.

Table 1 Types of manifolds used in this article.

	input shape	computation on	$\mathcal{S}$ in Equations 1 and 2
ECCobj2D	2D	2D: whole object	4-connected binary 2D shape
ECCobj	3D	3D: whole object	6-connected 3D voxel shape
ECCborder	3D	3D: border voxels	6 connected voxel surface in 3D, made out of voxels of the object that are 26 connected to a background voxel
ECCmesh	3D	2D: triangle mesh	connected triangle mesh of the surface of the 3D object

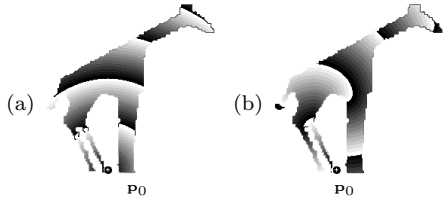

Fig. 1 Euclidean (a) and geodesic (b) distance function, for starting point $\mathbf{p}_0$. Gray values are distances modulo a constant.

Fig. 2 *ECC* of binary shape (point with smallest *ECC* marked).

Figure 1 shows a comparison of the geodesic and Euclidean distances. Figures 2 and 3 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 4 shows the difference between ECCobj and ECCborder, both using distances computed in the 3D volume.

3 Eccentricity Histogram Matching

To match two binary shapes/objects we first create a shape descriptor for each of them and then match these descriptors to obtain a similarity measure.

ECC histogram descriptor. The basic building block for our shape descriptor is the histogram $\mathbf{h}$ of the eccentricity transform ECC of the shape $\mathcal{S}$. We use k bins for the histogram. The histogram descriptor is the vector $\mathbf{h} \in \mathbb{R}^k$ defined by: $\forall i = 1, \ldots, k$

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

where $|\mathcal{S}|$ is the number of pixels/voxels in $\mathcal{S}$, and m and M are the smallest, respectively largest eccentricity values. The obtained histogram contains only bins for the values actually existing in the eccentricity transform i.e. from minimum to maximum eccentricity, and the sum of all bins is 1. Figures 5 and 6 and show examples of histograms for 2D and 3D shapes with different geometric features. We note that the histogram $\mathbf{h}$ is invariant under euclidean transformations, scaling and isometric bending of $\mathcal{S}$.

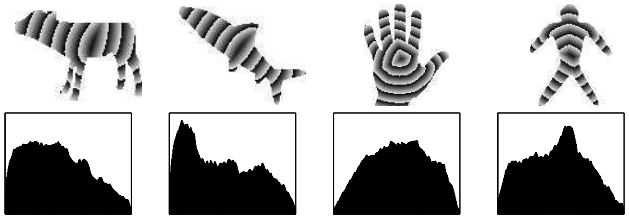

Fig. 5 Top: ECCobj2D for some 2D shapes. Bottom: corresponding histograms.

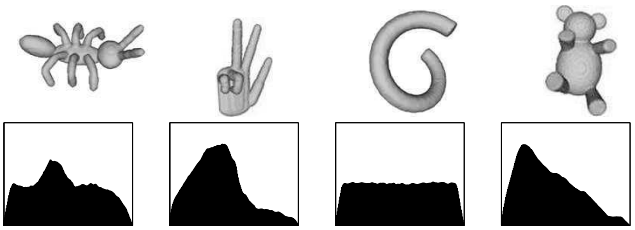

Fig. 6 Top: example 3D shapes. Bottom: corresponding ECCobj histograms.

Comparison of histograms. To match the descriptors of the two shapes $\mathcal{S}$ and $\tilde{\mathcal{S}}$, it is necessary to compute a meaningful distance between histograms. Let $\mathbf{h}, \tilde{\mathbf{h}} \in \mathbb{R}^k$ be the two histograms of $\mathcal{S}$ and $\tilde{\mathcal{S}}$ computed as above. We propose to use the simple ℓ^2-norm defined by

$$\delta(\mathbf{h}, \tilde{\mathbf{h}}) \stackrel{\text{def.}}{=} \sqrt{\sum_{i=1}^{k} (\mathbf{h}(i) - \tilde{\mathbf{h}}(i))^2}. \tag{3}$$

One could use more elaborate metrics such as the χ^2 metric or those defined in Osada et al (2002), but we found in numerical experiments that all these metrics give results similar to δ, which is the easiest and fastest to compute (discussion follows in Section 5).

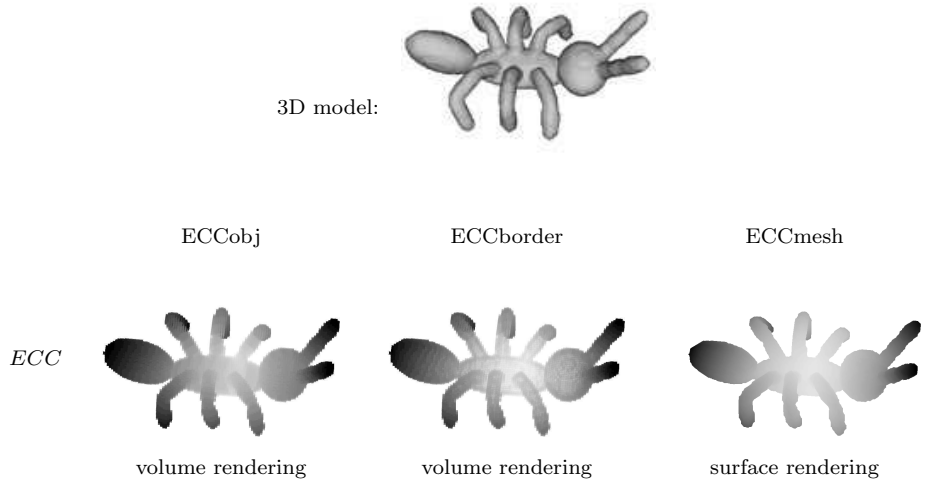

Fig. 3 Top: 3D model of an ant. Bottom: ECCobj, ECCborder, ECCmesh (darker = higher *ECC* value).

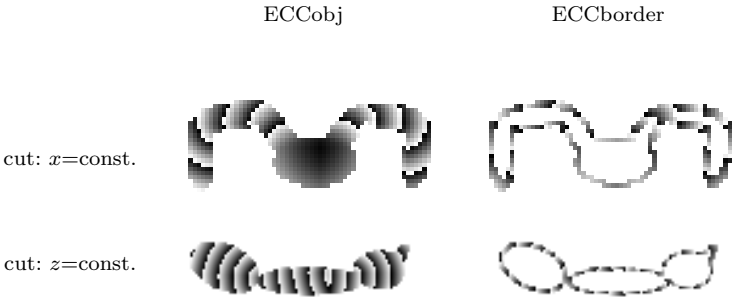

Fig. 4 Comparison between the two volume computations of *ECC*: ECCobj and ECCborder.

The dissimilarity $\Delta(\mathcal{S}, \tilde{\mathcal{S}})$ is computed between two shapes $\mathcal{S}$ and $\tilde{\mathcal{S}}$ as the distance of their histogram descriptors:

$$\Delta(S, \tilde{S}) \stackrel{\text{def.}}{=} \delta(\mathbf{h}, \tilde{\mathbf{h}}). \tag{4}$$

3.1 Characteristics of *ECC* Histograms.

The histogram of the *ECC* characterizes the compactness of the shape (e.g. a flat histogram characterizes a very elongated shape, a histogram with monotonically decreasing values characterizes a rather compact shape). Figure 7 shows example eccentricity histograms for basic shapes, with and without holes and articulation.

The histogram of the *ECC* of a simple open curve[4] $\mathcal{S}_a$ with length $l = d(\mathbf{e}_1, \mathbf{e}_2)$ (Figure 8(a)), is flat with a possibly smaller value in the first bin. The continuous formula is:

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

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

Consider $\mathcal{S}_b$ obtained by adding to $\mathcal{S}_a$ a simple open curve of length $d(\mathbf{q}_1, \mathbf{q}'_3) < l/2$ connected at the point $\mathbf{q}_1$ (Figure 8(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.

A possibility to change the histogram value for the center is to introduce cycles. Consider $\mathcal{S}_d$ obtained by adding a simple open curve of length $(\mathbf{q}_1, \mathbf{q}_2)$ to $\mathcal{S}_a$ (Figure 8(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

[4] In this article, the term *curve* is used to denote a one dimensional and continuous object, and includes both straight and non-straight lines.

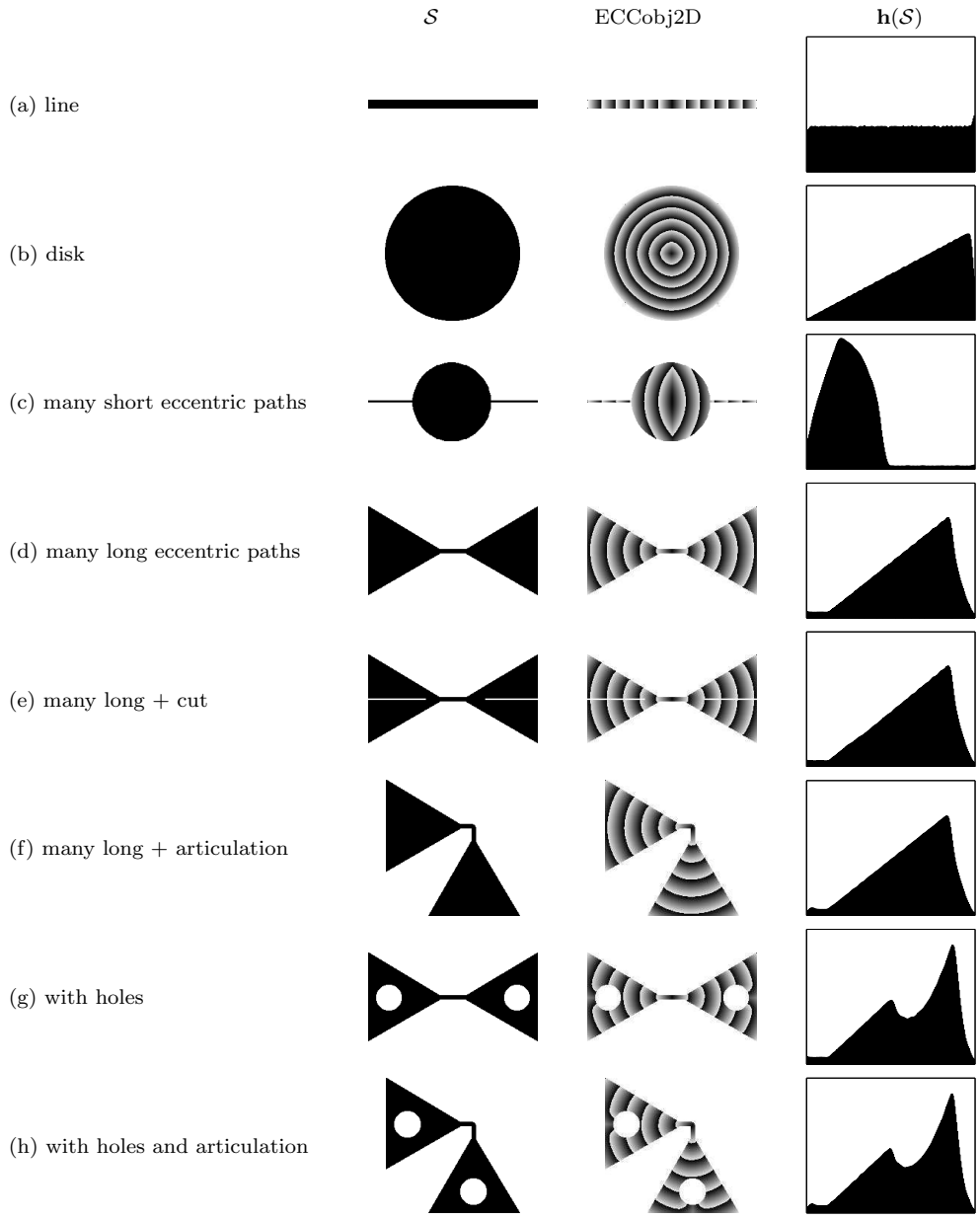

Fig. 7 Basic shapes and their eccentricity histograms.

$d(\mathbf{e}_1, \mathbf{c}) = d(\mathbf{e}_1, \mathbf{c}') = d(\mathbf{e}_1, \mathbf{e}_2)/2$. 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 are two additional points. If $\mathbf{q}_1 \to \mathbf{c}$, $\mathbf{q}_1 \neq \mathbf{c}$, only one additional point will exist for the eccentricity values $[d(\mathbf{c}, \mathbf{e}_1), d(\mathbf{q}_2, \mathbf{e}_1))$.

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[5], which limits the number of curves that can be put close to each other and not intersect. With the maximum shape size (number of pixels/voxels) and the number of bins k 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).

A histogram has a smaller dimension (1D) than the shape and a whole class of shapes is projected into the same histogram. Two shapes $\mathcal{S}$ and $\tilde{\mathcal{S}}$ with the same eccentricity histograms satisfy $\Delta(S, \tilde{S}) = 0$, and are thus considered to be the same according to our recognition algorithm. Consider $\mathcal{S}_c$ in Figure 8(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 $\mathcal{S}_c$. The two shapes $\mathcal{S}_b$ and $\mathcal{S}_c$ have the

[5] Depends on the discretization and maximum shape size.

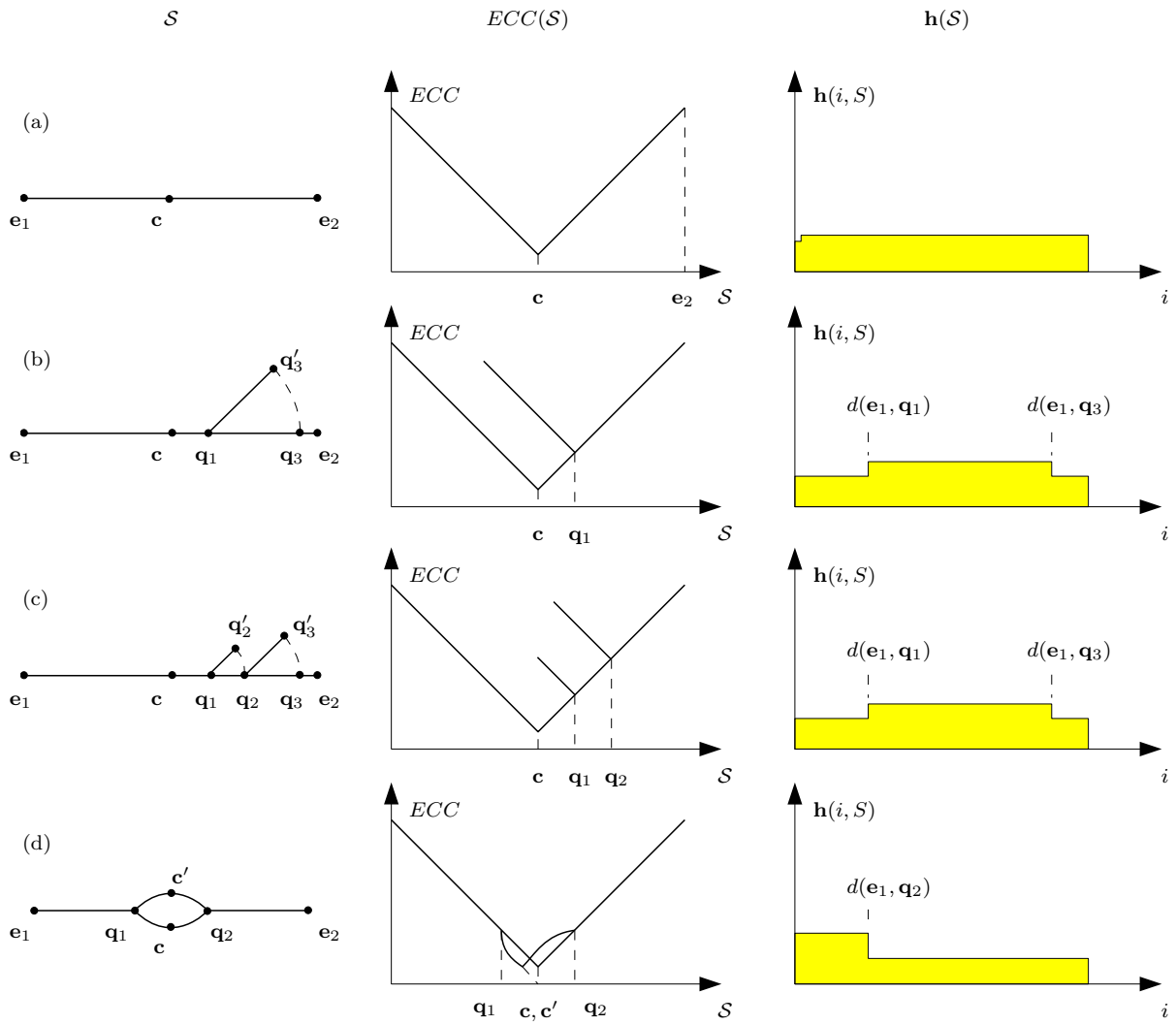

Fig. 8 Behavior of *ECC* histogram for basic changes in the shape. Column $\mathcal{S}$: where possible, straight lines where used for illustration, but only the length of the curves is relevant, not if straight or not.

same eccentricity histogram and cannot be differentiated using only that. One could say that eccentricity histograms are influenced by the structure of shapes (as new branches change the histogram), but they are not optimal for characterizing it.

On the other side, 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 (Figure 7 (g) and (h)), and gives good results comparable to many state of the art methods (Section 4).

4 Matching Experiments in 2D & 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 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 δ between two computed descriptors (ℓ^2-norm) has practically no CPU time consumption. A single, fixed-length vector as a descriptor can be a very efficient indexing method. The approaches compared with, are more complicated requiring decomposition of shapes, alignment/correspondences of features, etc.

4.1 2D Shape Matching

For the experiments we have used three shape databases: Kimia 25 (Sharvit et al, 1998), Kimia 99 (Sebastian et al, 2004) and MPEG7 CE-Shape-1 (Latecki et al, 2000).

A shape database is composed of q shapes $\{\mathcal{S}_i\}_{i=1}^{q}$ and each shape $\mathcal{S}_i$ has a label $L(i) \in \{1, \ldots, l_{\max}\}$. 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 three blocks of Figure 9 show the shapes from the Kimia 25 database, ordered by classes (such as fish, planes, rabbits, etc.). Any shape matching algorithm α assigns to each shape $\mathcal{S}_i$ a vector of best matches Φ_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, Φ_i contains all shapes including the query shape $\mathcal{S}_i$ (all 2D benchmarks in this article), or leave $\mathcal{S}_i$ out, i.e. the shape $\mathcal{S}_i$ is not matched to itself and Φ_i has $q-1$ elements (all 3D benchmarks in this article).

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

$$\text{Match}_k(\Phi) \stackrel{\text{def.}}{=} \sum_{i=1}^{q} 1_{L(\Phi_i(k))=L(i)} \leqslant q.$$

Tables 2 and 3 give the value of Match_k for various shape matching algorithms.

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:

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

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 20×1400 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 results of the presented approach over both Kimia 25 and Kimia 99, and over MPEG 7 are slightly bellow the state of the art.

Case Study - Kimia 25. Figure 9 shows the retrieval results for Kimia 25. The first column shows the 25 shapes $\mathcal{S}_i$. The following set of shapes forms an array, where the shape at row i and column k is $\Phi_i(k)$, the rank-k shape associated to $\mathcal{S}_i$.

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). The class with the best results are rabbits, followed by tools, hands, fishes, airplanes and greebles (the shapes that do not look like anything we know). 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 histograms of the greebles and the unoccluded hands (Figure 10). The histograms are very similar even though the shapes are of different classes, e.g. the histogram of the first greeble (Figure 10 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 our case, disregards certain structural properties of distances/paths (studied in detail in Section 3.1).

For the second question, consider the shapes in Figure 11 (a rabbit - $\mathcal{S}_{19}$, and two tools - $\mathcal{S}_{25}$ and $\mathcal{S}_{22}$), and the results, Φ_{25}, in row 25 of Figure 9. When matching $\mathcal{S}_{25}$, the rabbit has a better score than $\mathcal{S}_{22}$, even though one might say that the histograms of $\mathcal{S}_{25}$ and $\mathcal{S}_{22}$ reveal more similar distance characteristics than

Table 2 The value of $\text{Match}_k(\Phi)$ for various algorithms on the Kimia 25 database. See Ling and Jacobs (2007) for a description of these algorithms.

Algorithm α	k=1	2	3
Sharvit et. al	23	21	20
ECCobj2D	25	20	16
Gdalyahu and Weinshall	25	21	19
Belongie et. al	25	24	22
ID-Shape Context	25	24	25

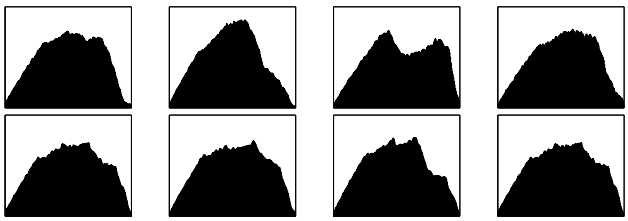

Fig. 10 Histograms for: top: greebles, bottom: unoccluded hands.

Table 4 The value of $\text{Bullseye}(\Phi)$ for various algorithms on the MPEG 7 databases.

Algorithm α	$\text{Bullseye}(\Phi)$
random	2.86%
ECCobj2D	44.28%
Shape Context	64.59%
ID-Shape Context	68.83%

Table 3 The value of $\text{Match}_k(\Phi)$ for various algorithms on the Kimia 99 database. See Ling and Jacobs (2007) for a description of these algorithms

Algorithm α	k=1	2	3	4	5	6	7	8	9	10
Shape Context	97	91	88	85	84	77	75	66	56	37
ECCobj2D	99	87	74	67	64	49	52	45	38	33
Gen. Model	99	97	99	98	96	96	94	83	75	48
Shock Edit	99	99	99	98	98	97	96	95	93	82
ID-Shape Context Belongie et al (2002)	99	99	99	98	98	97	97	98	94	79

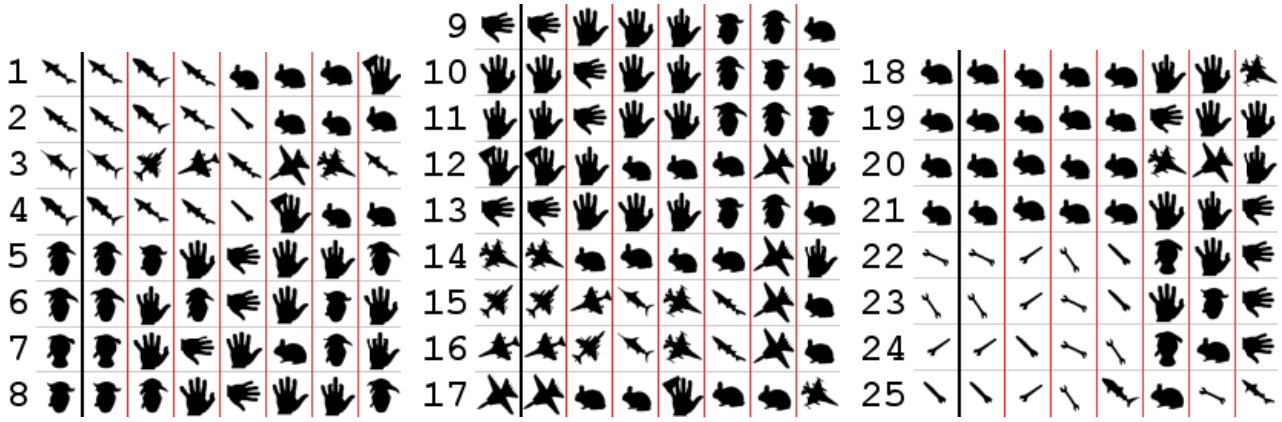

Fig. 9 Retrieval results for the single scale descriptor on the Kimia 25 database.

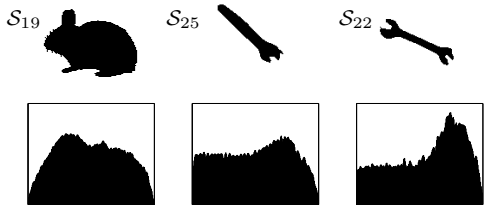

Fig. 11 Three shapes from the Kimia 25 database and their eccentricity histograms.

the histogram of $\mathcal{S}_{19}$ (see Figure 11). 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 is due to typical histogram matching methods, which are inherently low level and fail to capture the high level context of the task. Discussion follows in Section 5.

Geometrical properties of the shapes are well captured by our low-dimensional descriptors. For instance, elongated shapes are well separated from more compact shapes. However, more advanced geometrical features, such as intricate structural properties are thrown away by our signature extraction. This is for instance why the class 'greebles' is not separated enough from the class 'hands'.

4.2 3D Shape Matching

One of the most widely used 3D object retrieval databases is the Princeton Shape Benchmark (Shilane et al, 2004). 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 eccentricity is its robustness with respect to articulation, we have turned to the McGill Shape Benchmark (Zhang et al, 2005). 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. This article shows the results on the $q = 255$ shapes grouped into the $l_{\max} = 10$ classes of articulated shapes (Figure 12). Shapes are not matched to themselves and so Φ_i contains $q - 1$ shapes. See Siddiqi et al (2007) for results of other methods for the same benchmark.

Three ECC based descriptors where used (Figure 3):

1. **ECCobj** - eccentricity of the whole object;
2. **ECCborder** - eccentricity of the border voxels;
3. **ECCmesh** - eccentricity of the triangle mesh of the surface of the object.

ECCborder is obtained by computing $ECC(\partial^6\mathcal{S})$, where $\partial^6\mathcal{S}$ is the 6 connected voxel boundary of $\mathcal{S}$.

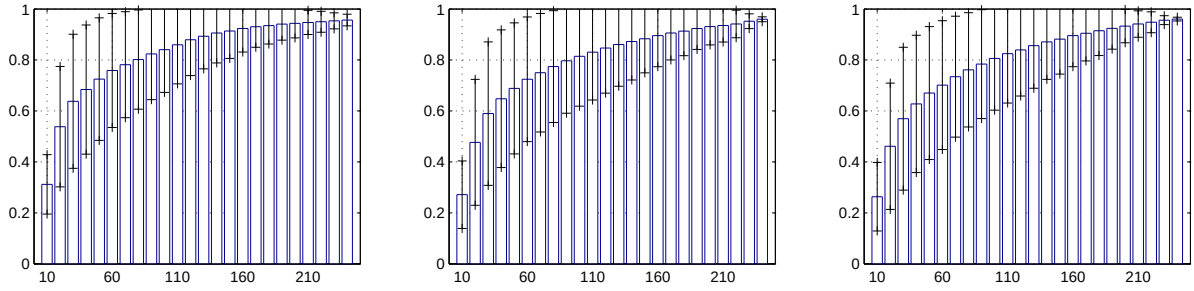

Fig. 13 Recall for several rank thresholds

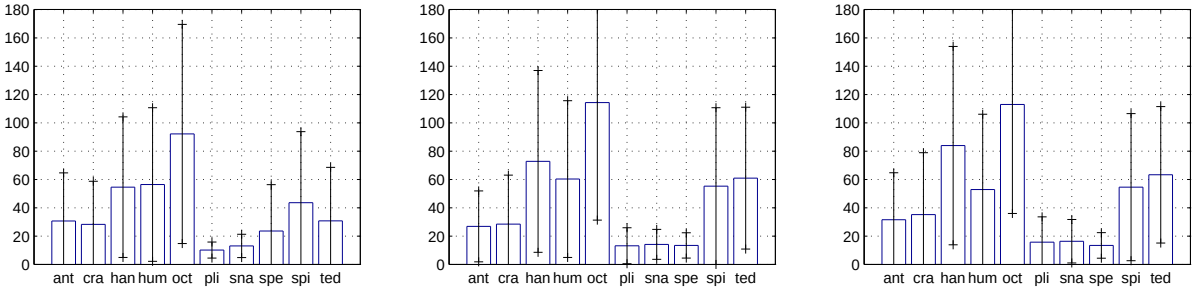

Fig. 14 Average ranks for each class. The first tree letters of each class name are printed.

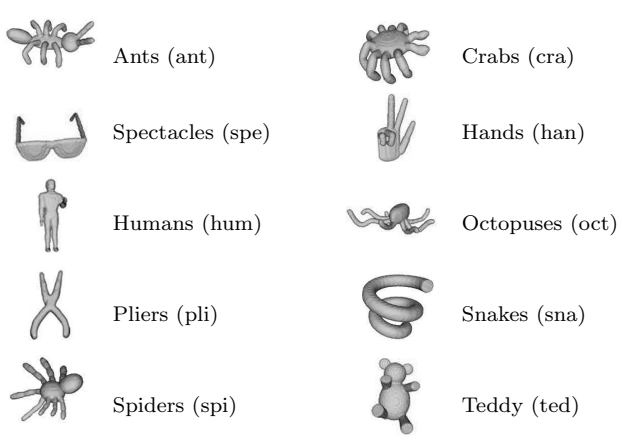

Fig. 12 The object classes from the McGill 3D shape database having significant part articulation.

ECCmesh is computed on the 2D manifold defined by the boundary of the 3D objects. 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 border do not have to be stored (e.g. interior of the object), 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 approximation volume.

Figure 13 shows the recall for several rank thresholds ($t = 10, 20, \dots$). The ratio of models in the database in the same category as the query, with indexing rank $\leq t$, to the total number of objects in the same category (never including the query itself) is given (see Section 4.1 for the notation):

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

The average results and standard deviation, over all classes, are given (Figure 13).

Figure 14 shows the average and the standard deviation of the ranks for each class (lower average is better). For all queries in a class, the average of the ranks of all other objects in that class are computed:

$$\mathrm{AvgRank}(\Phi_i) = \frac{1}{|\mathcal{Q}(L(i))| - 1} \sum_{k=1}^{q-1} k_{L(\Phi_i(k))=L(i)}.$$

Table 5 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.

Figure 15 shows the *precision-recall* curves for each of the 10 classes. 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 (Fawcett, 2006). Precision refers to the ratio of the relevant shapes retrieved, to the total number retrieved:

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

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.

As can be seen in Figures 13, 14, and 15, and Table 5, ECCobj does in most cases a better job than ECCborder and ECCmesh. The recall of the three methods is comparable, with slightly better results from ECCobj. With respect to the average ranks, ECCobj does better with the hands, octopus, pliers, snakes, spiders, teddy, is worse then one of ECCborder and ECCmesh with the ants, crabs, humans, and slightly worse than both other methods with the spectacles. None of the three variants produces an average class rank higher than 50%. All three methods have the smallest average class distance (highest similarity) correct for 8 out of 10 classes, with ECCobj having the correct class as the second smallest one for the other two, the humans and octopus (see Table 5).

Figure 15 shows comparative precision-recall results of ECCobj, ECCborder and ECCmesh, and three other methods:

- medial surfaces (MS) (Siddiqi et al, 2007);
- harmonic spheres (HS) (Kazhdan et al, 2003);
- shape distributions (SD) (Osada et al, 2002).

ECCobj, ECCborder, and ECCmesh are comparable, 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 ants, and HS and SD for teddy. For these best results the MS has the best precision-recall followed by the *ECC* based methods, followed by HS and SD. The worst results are achieved by ECCobj, ECCborder, and ECCmesh for the octopus, MS for the pliers, and HS and SD for the hands. The results of MS for the pliers are superior to ECCobj, ECCborder and ECCmesh for the octopus, which are in turn superior to the HS and SD for the hands. In comparison to all other three methods (MS, HS, SD), the eccentricity based methods score better on the pliers, spectacles and snakes.

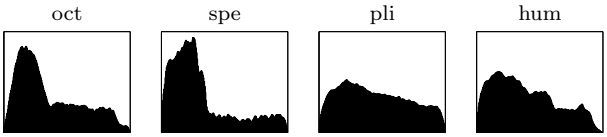

Fig. 16 Similar ECCobj histograms corresponding to 3D objects of different classes.

4.3 Discussion

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. In contrast, the most efficient shape matching algorithms (Ling and Jacobs, 2007; Siddiqi et al, 2007) are more complicated and require extraction of salient features and local signatures that need to be aligned or registered.

The major current limitations of our approach include: (1) Eccentricity histograms do not capture the topology 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.

As in the 2D case, connectivity of the isoheight lines/surfaces of the eccentricity transform does capture the part structure of a shape (Ion et al, 2008c), but the histograms 'throw away' this information. Figure 16 shows two pairs of similar histograms belonging to 3D objects of different classes.

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 looks more unstable. The eccentricity transform of a simply connected volume has in most of the cases a single stable center (minimum), while the eccentricity transform of its border will have a disconnected center or at least one with a more complex structure. The fact that ECCobj produced different results than ECCborder and ECCmesh for the teddy can be related to the compactness of the shapes and their parts.

Compared to other approaches (e.g. Siddiqi et al (2007)), one can identify the aspects discussed above (see Figure 15 and Table 5). For classes with simple topology (e.g. snakes and spectacles), the results are very good. For classes where part decomposition and structure play an important role (e.g. octopus v.s. spiders and crabs), the discrimination capabilities are reduced.

Table 5 Average matching results multiplied by 100 (smaller means more similar). For each row, the first and second smallest value are printed in bold.

ECCobj	ants	crabs	hands	humans	octopus	pliers	snakes	spectacles	spiders	teddy
ants	**1.75**	5.33	3.72	3.53	7.20	2.95	**2.91**	7.25	5.43	3.72
crabs	5.33	**1.55**	3.73	3.50	3.67	3.02	4.36	3.99	3.43	**2.53**
hands	3.72	3.73	**2.30**	3.04	5.60	2.71	3.76	6.19	4.53	**2.51**
humans	3.53	3.50	3.04	**2.19**	5.03	**2.13**	3.15	5.04	3.52	2.62
octopus	7.20	**3.67**	5.60	5.03	**3.90**	5.02	6.22	4.04	4.06	4.58
pliers	2.95	3.02	2.71	2.13	5.02	**0.55**	**1.82**	4.97	3.64	2.11
snakes	2.91	4.36	3.76	3.15	6.22	**1.82**	**0.80**	5.73	4.83	3.55
spectacles	7.25	3.99	6.19	5.04	4.04	4.97	5.73	**2.24**	**3.97**	5.13
spiders	5.43	**3.43**	4.53	3.52	4.06	3.64	4.83	3.97	**2.25**	3.50
teddy	3.72	2.53	2.51	2.62	4.58	**2.11**	3.55	5.13	3.50	**1.46**

ECCborder	ants	crabs	hands	humans	octopus	pliers	snakes	spectacles	spiders	teddy
ants	**1.00**	2.45	2.16	**1.60**	3.09	1.61	2.42	5.62	2.47	1.66
crabs	2.45	**1.41**	**2.30**	3.01	3.42	2.88	3.64	6.92	3.08	2.38
hands	**2.16**	2.30	**1.94**	2.65	2.98	2.48	3.49	6.05	2.78	2.29
humans	1.60	3.01	2.65	**1.57**	3.19	**1.54**	2.16	5.12	2.54	1.93
octopus	3.09	3.42	2.98	3.19	2.97	**2.72**	3.82	4.80	**2.69**	2.80
pliers	1.61	2.88	2.48	1.54	2.72	**0.65**	1.75	4.51	2.00	**1.47**
snakes	2.42	3.64	3.49	2.16	3.82	**1.75**	**0.85**	4.86	3.21	2.44
spectacles	5.62	6.92	6.05	5.12	4.80	**4.51**	4.86	**1.67**	4.69	5.26
spiders	2.47	3.08	2.78	2.54	2.69	**2.00**	3.21	4.69	**1.76**	2.07
teddy	1.66	2.38	2.29	1.93	2.80	**1.47**	2.44	5.26	2.07	**1.45**

ECCmesh	ants	crabs	hands	humans	octopus	pliers	snakes	spectacles	spiders	teddy
ants	**0.97**	2.34	1.94	1.66	2.46	**1.40**	2.36	4.78	1.93	1.47
crabs	2.34	**1.47**	2.52	3.05	3.09	2.88	3.67	6.41	2.75	**2.34**
hands	**1.94**	2.52	**1.98**	2.69	2.62	2.34	3.40	5.30	2.40	2.17
humans	1.66	3.05	2.69	**1.48**	3.03	**1.60**	1.91	4.56	2.48	2.07
octopus	2.46	3.09	2.62	3.03	2.61	2.39	3.54	4.65	**2.33**	**2.34**
pliers	**1.40**	2.88	2.34	1.60	2.39	**0.70**	1.78	3.92	1.71	1.44
snakes	2.36	3.67	3.40	1.91	3.54	**1.78**	**0.98**	4.00	2.95	2.56
spectacles	4.78	6.41	5.30	4.56	4.65	**3.92**	4.00	**1.66**	4.49	4.71
spiders	1.93	2.75	2.40	2.48	2.33	1.71	2.95	4.49	**1.50**	**1.68**
teddy	1.47	2.34	2.17	2.07	2.34	**1.44**	2.56	4.71	1.68	**1.37**

5 Parameters and Improvements

The number of bins for the histogram: The approach has one parameter, the number of bins k, of the histograms $\mathbf{h}$. In experiments, we used $k = 200$, which was chosen based on a few initial trials on a smaller set of shapes.

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 ($\max(\mathbf{h}^c) = \max(ECC)$). 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 has to satisfy:

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

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

As k 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*, 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

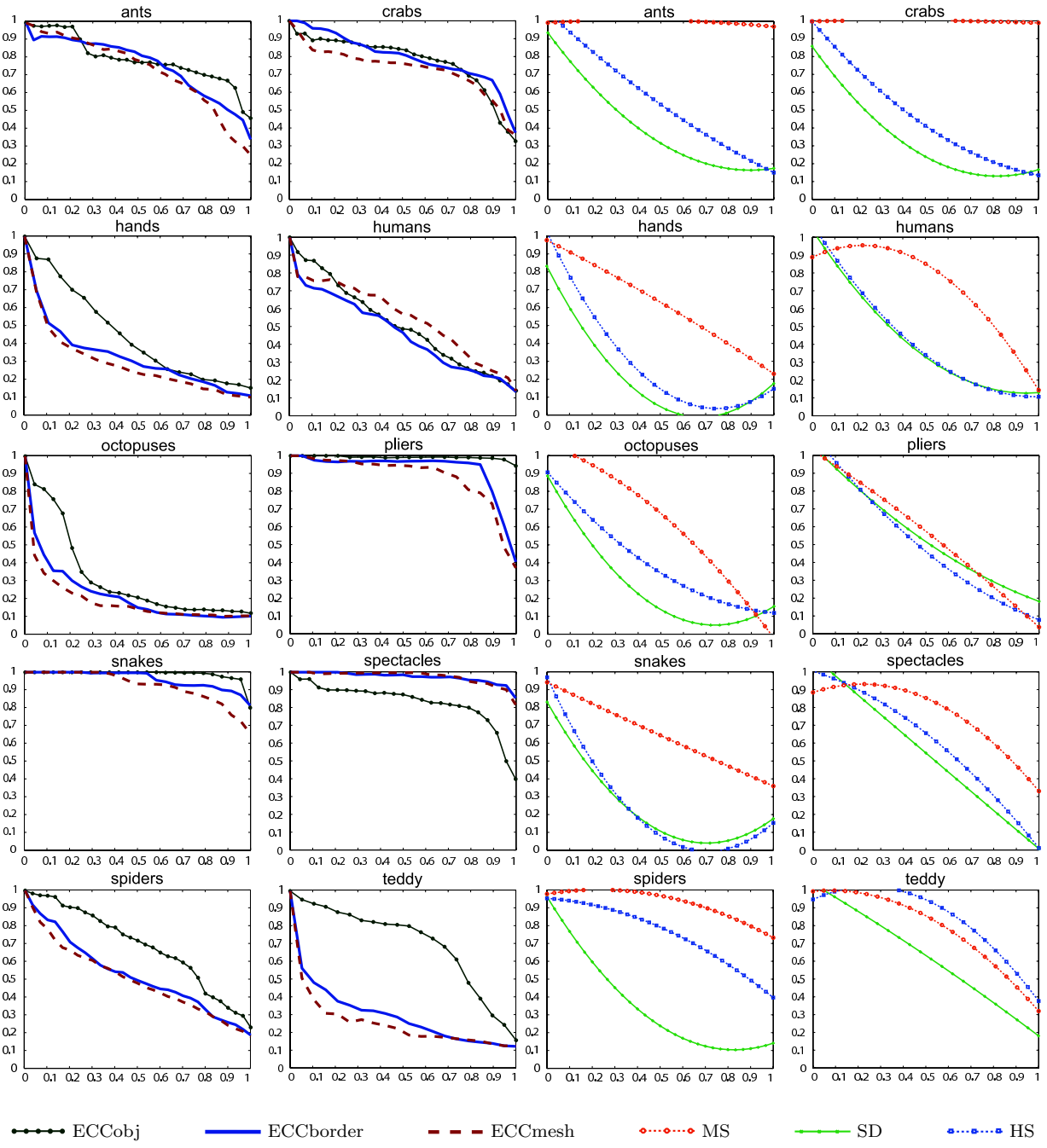

Fig. 15 Precision-recall for the ten classes. Left two columns: ECCobj, ECCborder, ECCmesh. Right two columns (image taken from Siddiqi et al (2007), with kind permission of Springer Science and Business Media): results of three other methods on the same database: medial surfaces (MS) (Siddiqi et al, 2007), harmonic spheres (HS) (Kazhdan et al, 2003), and shape distributions (SD) (Osada et al, 2002). Precision: horizontal axis, recall: vertical axis. (Best visualized in color.)

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 distance (equilateral triangle), and this number increases to 4 in 3D (regular tetrahedron).

Assuming that shapes from the same class have similar histograms, given the number of classes (vertices) and the required relations (weighted edges), a lower bound for the number of bins is equal to the smallest dimension in which the classes can be embedded s.t.

the weights of the edges corresponding to the distance between the vertices. If the variation inside classes increases the number of classes that can be discriminated will decrease.

Describing topology: One of the problems identified in Section 3.1 and during the experiments (Sections 4.1 and 4.2) is that the histograms do not capture the exact structure of the shape. Classical methods to describe the topology of a shape (e.g. Reeb graphs, Reeb (1964), and homology generators, Munkres (1993)) fail to capture the geometrical aspects. An approach to deal with this problem is presented in Aouada et al (2008). To describe a shape, two descriptors are used: a geometric one, based on the *Global Geodesic Function*(GCF), 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 GCF as the Morse function.

Initial steps in combining the eccentricity transform with Reeb graphs have been presented in Ion et al (2008c).

A better histogram matching: The problem of having a descriptor matching function that is aware of the context in which it is applied can be approached in two ways: use expert knowledge about the context to create an algorithm that considers the proper features, or learn the important features by giving a set of representative examples. In Yang and Jin (2006); Yang et al (2006), 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 eccentricity histogram distance computation as given by the training examples.

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 be able to discriminate between 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 novel 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, Salt and Pepper noise, and robust with respect to minor segmentation errors. Descriptors are normalized histograms of the eccentricity transform, compact, and easy to match. The method is straight-forward but still efficient, with experimental results comparable to more complex state of the art methods. Experimental results on popular 2D and 3D shape matching benchmarks are given, with computation on binary 2D images, binary 3D voxel objects, and 3D triangle meshes. The experiments are preceded by a detailed analysis of the properties of the descriptor and followed by in depth discussion of results, parameters, and improvement possibilities.

Appendix

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

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

for all $\mathbf{q} \in \mathcal{S}, ECC(\mathbf{q}) \leftarrow 0$ /**initialize distance matrix**/
$\mathbf{p} \leftarrow$ random point of $\mathcal{S}$ /**find a starting point**/

/**Phase 1: find a diameter**/
while $\mathbf{p}$ not computed **do**
  $ECC \leftarrow \max\{ECC, D^{\mathcal{S}}(\mathbf{p})\}$ /**accumulate & mark* $\mathbf{p}$ *as computed**/
  $\mathbf{p} \leftarrow \arg\max\{ECC(\mathbf{p}) | \mathbf{p} \in \mathcal{S}\}$ /**highest current ECC (farthest away)**/
end while

/**Phase 2: find center points and local maxima**/
$pECC \leftarrow 0$ /**make sure we enter the loop**/
while $pECC \neq ECC$ **do**
  $pECC \leftarrow ECC$
  $C \leftarrow \arg\min\{ECC(\mathbf{p}) | \mathbf{p} \in \mathcal{S}\}$ /**points with min. ECC**/
  for all $\mathbf{c} \in C$, $\mathbf{c}$ not computed **do**
    $D \leftarrow D^{\mathcal{S}}(\mathbf{c})$ /**compute distances from the center**/
    $ECC \leftarrow \max\{ECC, D\}$ /**accumulate & mark* $\mathbf{c}$ *as computed**/

    $\mathcal{M} \leftarrow \{\mathbf{q} \in \mathcal{S} \mid D(\mathbf{q})$ local max. in $\mathcal{S}$ & $\mathbf{q}$ not computed$\}$
    for all $\mathbf{m} \in \mathcal{M}$, $\mathbf{m}$ not computed **do**
      $ECC \leftarrow \max\{ECC, D^{\mathcal{S}}(\mathbf{m})\}$ /**accumulate & mark* $\mathbf{m}$ *as computed**/
    end for
  end for
end while

Output: Distances ECC.

For completeness, the algorithm *ECC06* (Kropatsch et al, 2006) used to compute the eccentricity transform

for the shapes in our experiments is included (see Ion et al (2008b) for an analysis of the speed/error performance). *ECC06* (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

Ankerst M, Kastenmüller G, Kriegel HP, Seidl T (1999) 3d shape histograms for similarity search and classification in spatial databases. In: 6th International Symposium on Advances in Spatial Databases, Springer, London, UK, pp 201–226

Ansary T, Vandeborre JP, Mahmoudi S, Daoudi M (2004) A bayesian framework for 3d models retrieval based on characteristic views. 2nd International Symposium on 3D Data Processing, Visualization and Transmission, 2004 3DPVT 2004 pp 139–146

Aouada D, Dreisigmeyer DW, Krim H (2008) 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

Belongie S, Malik J, Puzicha J (2002) Shape matching and object recognition using shape contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(4):509–522

Bronstein AM, Bronstein MM, Bruckstein AM, Kimmel R (2006) Matching two-dimensional articulated shapes using generalized multidimensional scaling. In: Conference on Articulated Motion and Deformable Objects, Springer, Mallorca, Spain, LNCS, vol 4069, pp 48–57

Bustos B, Keim DA, Saupe D, Schreck T, Vranić DV (2005) Feature-based similarity search in 3d object databases. ACM Comput Surv 37(4):345–387

Chen DY, Ouhyoung M, Tian XP, Shen YT, Ouhyoung M (2003) On visual similarity based 3d model retrieval. In: Eurographics, Granada, Spain, pp 223–232

Cyr CM, Kimia BB (2004) A similarity-based aspect-graph approach to 3d object recognition. International Journal of Computer Vision 57(1):5–22

Diestel R (1997) Graph Theory. Springer, New York

Elad M, Tal A, Ar S (2001) Content based retrieval of vrml objects: an iterative and interactive approach. In: Eurographics Workshop on Multimedia, Springer, New York, pp 107–118

Fawcett T (2006) An introduction to roc analysis. Pattern Recognition Letters 27(8):861–874

Gorelick L, Galun M, Sharon E, Basri R, Brandt A (2004) Shape representation and classification using the poisson equation. In: CVPR (2), pp 61–67

Hamza AB, Krim H (2003) Geodesic object representation and recognition. In: DGCI: International Workshop on Discrete Geometry for Computer Imagery

Harary F (1969) Graph Theory. Addison Wesley

Hilaga M, Shinagawa Y, Kohmura T, Kunii TL (2001) 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, pp 203–212

Ion A, Peyré G, Haxhimusa Y, Peltier S, Kropatsch WG, Cohen L (2007) Shape matching using the geodesic eccentricity transform - a study. In: W Ponweiser CB M Vincze (ed) The 31st Annual Workshop of the Austrian Association for Pattern Recognition (OAGM/AAPR), OCG, Schloss Krumbach, Austria, pp 97–104

Ion A, Artner NM, Peyré G, López Mármol SB, Kropatsch WG, Cohen L (2008a) 3d shape matching by geodesic eccentricity. In: Workshop on Search in 3D (in conjunction with CVPR 2008), IEEE, Anchorage, Alaska

Ion A, Kropatsch WG, Andres E (2008b) Euclidean eccentricity transform by discrete arc paving. In: Coeurjolly D, Sivignon I, Tougne L, Dupont F (eds) 14th IAPR International Conference on Discrete Geometry for Computer Imagery (DGCI), Springer, Lyon, France, Lecture Notes in Computer Science, vol LNCS 4992, pp 213–224

Ion A, Peltier S, Alayrangues S, Kropatsch WG (2008c) Eccentricity based topological feature extraction. In: Alayrangues S, Damiand G, Fuchts L, Lienhardt P (eds) Workshop on Computational Topology in Image Context, Poitiers, France

Ip CY, Regli L W C Sieger, Shokoufandeh A (2003) Automated learning of model classifications. In: 8th ACM Symposium on Solid Modeling and Applications, ACM Press, New York, pp 322–327

Johnson A, Hebert M (May 1999) Using spin images for efficient object recognition in cluttered 3d scenes.

IEEE Transactions on Pattern Analysis and Machine Intelligence 21(5):433–449

Kazhdan M, Funkhouser T, Rusinkiewicz S (2003) 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, pp 156–164

Klette R, Rosenfeld A (2004) Digital Geometry. Morgan Kaufmann

Kropatsch WG, Ion A, Haxhimusa Y, Flanitzer T (2006) The eccentricity transform (of a digital shape). In: 13th International Conference on Discrete Geometry for Computer Imagery (DGCI), Springer, Szeged, Hungary, pp 437–448

Latecki LJ, Lakämper R, Eckhardt U (2000) 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, pp 1424–1429

Ling H, Jacobs DW (2007) Shape classification using the inner-distance. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(2):286–299

Maisonneuve F, Schmitt M (1989) An efficient algorithm to compute the hexagonal and dodecagonal propagation function. Acta Stereologica 8(2):515–520

Mokhtarian F, Mackworth AK (1992) A theory of multiscale, curvature-based shape representation for planar curves. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(8):789–805

Munkres JR (1993) Elements of Algebraic Topology. Addison-Wesley

Osada R, Funkhouser T, Chazelle B, Dobkin D (2002) Shape distributions. ACM Transactions on Graphics 21(4):807–832

Paquet E, Rioux M (1999) Nefertiti: A tool for 3-d shape databases management. SAE transactions 108:387–393

Paquet E, Murching A, Naveen T, Tabatabai A, Rioux M (2000) Description of shape information for 2-d and 3-d objects. Signal Processing: Image Communication 16(1–2):103–122

Reeb G (1964) 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):37–42

Reuter M, Wolter FE, Peinecke N (2005) Laplace-spectra as fingerprints for shape matching. In: Kobbelt L, Shapiro V (eds) Symposium on Solid and Physical Modeling, ACM, pp 101–106

Schmitt M (1993) Propagation function: Towards constant time algorithms. Acta Stereologica: Proceedings of the 6th European Congress for Stereology, September 7-10, Prague 13(2)

Sebastian TB, Klein PN, Kimia BB (2004) Recognition of shapes by editing their shock graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(5):550–571

Sethian J (1999) Level Sets Methods and Fast Marching Methods, 2nd edn. Cambridge Univ. Press

Sharvit D, Chan J, Tek H, Kimia B (1998) Symmetry-based indexing of image databases. In: IEEE Workshop on Content-based Access of Image and Video Libraries, pp 56–62

Shilane P, Min P, Kazhdan MM, Funkhouser TA (2004) The princeton shape benchmark. In: International Conference on Shape Modeling and Applications (SMI), Genova, Italy, IEEE Computer Society, pp 167–178

Shinagawa Y, Kunii T, Kergosien Y (1991) Surface coding based on morse theory. Computer Graphics and Applications, IEEE 11(5):66–78

Siddiqi K, Shokoufandeh A, Dickinson S, Zucker SW (1999) Shock graphs and shape matching. International Journal of Computer Vision 30:1–24

Siddiqi K, Zhang J, Macrini D, Shokoufandeh A, Bouix S, Chen R, Dickinson S (2007) Retrieving articulated 3-d models using medial surfaces. Machine Vision and Applications

Soille P (2002) Morphological Image Analysis, 2nd edn. Springer

Sundar H, Silver D, Gagvani N, Dickinson S (2003) Skeleton based shape matching and retrieval. Shape Modeling International, 2003 pp 130–139

Suri S (1987) The all-geodesic-furthest neighbor problem for simple polygons. In: Symposium on Computational Geometry, pp 64–75

Tangelder JWH, Veltkamp RC (2003) Polyhedral model retrieval using weighted point sets. In: Shape Modeling International, IEEE, Seoul, Korea, pp 119–129

Veltkamp RC, Latecki L (2006) Properties and performance of shape similarity measures. In: Proceedings of the 10th IFCS Conference on Data Science and Classification, Slovenia

Vranic D, Saupe D (2001a) 3d model retrieval with spherical harmonics and moments. In: 23rd DAGM-Symposium on Pattern Recognition, Springer, London, UK, pp 392–397

Vranic D, Saupe D (2001b) 3d shape descriptor based on 3d fourier transform. In: EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services, Comenius University, pp 271–274

Vranic D, Saupe D (2002) Description of 3d-shape using a complex function on the sphere. In: International Conference on Multimedia and Expo, IEEE, pp 177–

180

Yang L, Jin R (2006) 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

Yang L, Jin R, Sukthankar R, Liu Y (2006) 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

Zaharia T, Preux F (2001) Three-dimensional shape-based retrieval within the mpeg-7 framework. In: SPIE Conference on Nonlinear Image Processing and Pattern Analysis XII, pp 133–145

Zahn CT, Roskies RZ (1972) Fourier descriptors for plane closed curves. IEEE Transactions on Computer 21(3):269–281

Zhang J, Siddiqi K, Macrini D, Shokoufandeh A, Dickinson SJ (2005) 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, Springer, LNCS, vol 3757, pp 285–300