3D Shape Matching by Geodesic Eccentricity
Adrian Ion, Nicole M. Artner, Gabriel Peyré, Salvador B. Lopez Marmol, Walter G. Kropatsch, Laurent D. Cohen

To cite this version:

HAL Id: hal-00419826
https://hal.archives-ouvertes.fr/hal-00419826
Submitted on 25 Sep 2009

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.
3D Shape Matching by Geodesic Eccentricity*

Adrian Ion
PRIP, Vienna University of Technology
ion@prip.tuwien.ac.at

Gabriel Peyré
CEREMADE, Université Paris-Dauphine
peyre@ceremade.dauphine.fr

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

Salvador B. López Már mol
PRIP, Vienna University of Technology
salva@prip.tuwien.ac.at

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

Laurent Cohen
CEREMADE, Université Paris-Dauphine
cohen@ceremade.dauphine.fr

Abstract

This paper makes use of the continuous eccentricity transform to perform 3D shape matching. The eccentricity transform has already been proved useful in a discrete graph-theoretic setting and has been applied to 2D shape matching. We show how these ideas extend to higher dimensions. The eccentricity transform is used to compute descriptors for 3D shapes. These descriptors are defined as histograms of the eccentricity transform and are naturally invariant to euclidean motion and articulation. They show promising results for shape discrimination.

1. Introduction

Shape recognition is a central topic in computer vision. It requires to set up a signature that characterizes the properties of interest for the recognition [27]. The invariance of the signature or the matching method to changes in orientation and scale, and to local deformations such as articulation is important for the identification of 2D and 3D shapes.

Applications of shape recognition can be found in Computer Aided Design/Computer Aided Manufacturing (CAD/CAM), virtual reality (VR), medicine, molecular biology, security, and entertainment [3].

Existing approaches can be divided into [3]: Statistical descriptors, like for example geometric 3D moments employed by [6, 19] and the shape distribution [17, 12]. Extension-based descriptors, which are calculated from features sampled along certain directions from a position within the object [30, 28]. Volume-based descriptors use the volumetric representation of a 3D object to extract features (examples are Shape histograms [1], Model Voxelization [29] and point set methods [26]). Descriptors using the surface geometry compute curvature measures and/or the distribution of surface normal vectors [20, 32]. Image-based descriptors reduce the problem of 3D shape matching to an image similarity problem by comparing 2D projections of the 3D objects [2, 5, 4]. Methods matching the topology of the two objects (for example Reeb graphs, where the topology of the 3D object is described by a graph structure [8, 23]). 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 [25]. 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 [13].

The majority of shape descriptors is quite complex and not invariant to the deformation or articulation of object parts. In this article we extend the simple and efficient approach to shape matching with the eccentricity transform [11] to 3D. Based on the computation of geodesic lengths, eccentricity histograms are robust with respect to changes in orientation, scale and articulation of the object. The presented approach could be fitted to either of the categories extension-based or volume-based. In Section 2 the eccentricity transform is recalled. Section 3 presents the eccentricity histogram descriptors. Experimental results and discussion follow in Section 4.

* Partially supported by the Austrian Science Fund under grants S9103-N13 and P18716-N13.
2. Eccentricity Transform

The following definitions and properties follow [15, 10], and are extended to \( n \)-dimensional domains.

Let the shape \( S \) be a closed set in \( \mathbb{R}^n \). A (geodesic) path \( \pi \) is the continuous mapping from the interval \([0,1]\) to \( S \). Let \( \Pi(p_1,p_2) \) be the set of all paths between two points \( p_1, p_2 \in S \) within the set \( S \). The geodesic distance \( d(p_1,p_2) \) between two points \( p_1, p_2 \in S \) is defined as the length \( \lambda(\pi) \) of the shortest path \( \pi \in \Pi(p_1,p_2) \) between \( p_1 \) and \( p_2 \).

\[
d(p_1,p_2) = \min \{ \lambda(\pi(p_1,p_2)) | \pi \in \Pi(p_1,p_2) \} \tag{1}
\]

where

\[
\lambda(\pi(t)) = \int_0^1 |\dot{\pi}(t)| dt
\]

and \( \pi(t) \) is a parametrization of the path from \( p_1 = \pi(0) \) to \( p_2 = \pi(1) \). The eccentricity transform of \( S \) can be defined as, \( \forall p \in S \)

\[
ECC(S,p) = \max \{ d(p,q) | q \in S \}, \tag{2}
\]

i.e. to each point \( p \) it assigns the length of the shortest geodesics to the points farthest away from it.

In [15] it is shown that this transformation is quasi-invariant to articulated motion and robust against salt and pepper noise (which creates holes in the shape). An analysis of the variation of geodesic distance under articulation can be found in [16].

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). A definition for graphs is given in [15].

This paper considers the class of 6-connected discrete shapes \( S \) defined by points on a square grid \( \mathbb{Z}^2 \). Paths need to be contained in the area of \( \mathbb{R}^3 \) defined by the union of the support cubes for the voxels of \( S \). The distance between any two voxels, whose connecting segment is contained in \( S \), is computed using the \( \ell_2 \)-norm. (For easy understanding, some images with 2D shapes are also shown). The eccentricity transform is computed for:

- the whole object, i.e. \( S \) in Equations 1 and 2 is the 3D object itself (all voxels of the object);
- the border voxels of the object, i.e. \( S \) in Equations 1 and 2 is the 6 connected voxel surface in 3D, made out of voxels of the object that are 26 connected to a background voxel.

An eccentric point is a point that reaches a maximum in Equation 2, and all eccentric points lie on the border of \( S \) [15]. The center is the set of points that have the smallest eccentricity.

2.1. Computation

The straight forward approach is: for each point of \( S \), compute the distance to all other points and take the maximum. In [10] faster computation and efficient approximation algorithms are presented. For this paper, the fastest one, algorithm ECC06, has been used (see [10] for an analysis of the speed/error performance).

ECC06 relies on the computation of the shape bounded single source distance transform \( DT(S,p) \) (Figure 1.b), which is computed for estimated eccentric point candidates in an iterative manner. The shape bounded single source distance transform \( DT(S,p) \) associates to each point \( q \in S \) the geodesic distance to \( p \). The 3D \( DT \) can be computed using Fast Marching [21] which allows for an efficient computation in \( O(n \log(n)) \) steps for a grid with \( n \) points.

Figure 1 shows a comparison of the geodesic and Euclidean distances. Figures 2 and 3 show the eccentricity transform of two shapes. For the case of the 3D shape, the eccentricity transform is presented for the whole object and for the border voxels.

3. Eccentricity Histogram Matching

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

\[\text{Figure 1. An example of (a) Euclidean and (b) geodesic distance function, for some starting point } p_0.\]

\[\text{Figure 2. Left: a 2D binary shape. Right: eccentricity transform } ECC \text{ - same gray value assignment as in Figure 1 (the point with smallest eccentricity is marked).}\]
3D model:

<table>
<thead>
<tr>
<th>I</th>
<th>II</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>(a)</td>
</tr>
<tr>
<td>(b)</td>
<td>(b)</td>
</tr>
<tr>
<td>(c)</td>
<td>(c)</td>
</tr>
</tbody>
</table>

Figure 3. Top: 3D model of an ant. Column I: eccentricity transform of the ant. Column II: eccentricity transform of the border voxels of the ant. (a) Volume visualization of the eccentricity transform - darker means higher eccentricity, left and right use different gray value scales. (b) Cut along plane \( x = \text{const} \). (c) Cut along plane \( z = \text{const} \). (a) and (b) use the same gray value assignment as in Figure 1.

approach for using ECC histograms to match 2D Shapes can be found in [11].

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

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

where \(|S|\) is the number of pixels in \( S \), 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. Fig. 4 shows examples of histograms for shapes with different geometric features. We note that the histogram \( \mathbf{h} \) is invariant under euclidean transformations, scaling and isometric bending of \( S \). In numerical applications, we use \( k = 200 \) and each descriptor is normalized such that the sum of all bins is one.

\[
\delta(\mathbf{h}, \tilde{\mathbf{h}}) \overset{\text{def.}}{=} \sqrt{\sum_{i=1}^{k} (h(i) - \tilde{h}(i))^2}.
\]

One could use more elaborate metrics such as the \( \chi^2 \) metric or those defined in [18], but we found in numerical experiments that all these metrics give results similar to \( \delta \), which is the easiest to compute.

We can compute the distance \( \Delta(S, \tilde{S}) \) between two shapes \( S \) and \( \tilde{S} \) as the distance of their histogram descriptors

\[
\Delta(S, \tilde{S}) \overset{\text{def.}}{=} \delta(\mathbf{h}, \tilde{\mathbf{h}}).
\]

**Comparison of histograms.** In order to match the descriptors of the two shapes \( S \) and \( \tilde{S} \), we need to compute a meaningful distance between histograms. Let \( \mathbf{h}, \tilde{\mathbf{h}} \in \mathbb{R}^k \) be the two histograms of \( S \) and \( \tilde{S} \) computed as above. We propose to use the simple \( \ell_2 \)-norm defined by

\[
\Delta(S, \tilde{S}) \overset{\text{def.}}{=} \delta(\mathbf{h}, \tilde{\mathbf{h}}).
\]

4. Experiments

One of the most widely used 3D object retrieval databases is the Princeton Shape Benchmark [22]. 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 objects 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 [33]. 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 objects, 255 have significant part articulation. These articulated objects are grouped into 10 classes (see Figure 5). This paper shows the results on the 10 enumerated classes. See [24] for results of other methods for the same benchmark.

Two descriptors were used (see Figure 3):

1. **ECCobj** - eccentricity of the whole object;
2. **ECCborder** - eccentricity of the border voxels.
Figure 5. The object classes from the McGill 3D shape database having significant part articulation.

Figure 6. Percentage recall for several rank thresholds. Top: ECCobj, Bottom: ECCborder.

ECCborder is an attempt to get a feeling for how the eccentricity transform of the boundary relates to the eccentricity transform of the object. In the ideal case, the former should be computed on the 2D manifold defined by the boundary of the 3D objects. Computing in the “volume” of the 6 connected voxel boundary is an approximation that is possible with the available tools.

Figure 6 shows the percentage recall for several rank thresholds ($t = 10, 20, \ldots$). For each rank $t$, the percentage of models in the database in the same category as the query (not including the query itself) with indexing rank $\leq t$ is given. The average results and standard deviation, over all classes, are given.

Figure 7 shows the average ranks for each class. For all queries in a class, the ranks of all other objects in that class are computed. The average and standard deviation for each class are given.

Tables 1 and 2 show the average score (see Equation 3) for all pairs of classes. Each object in the database is matched against all other objects and each cell shows the average score between all combinations of objects of the two classes defined by the row and column. See Section 4.1 for a discussion on the results.

Figure 8 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 [7].

Given a query object $Q$ and retrieval result $R$ contain-
ing $t$ objects, precision refers to the ratio of the relevant objects retrieved to the total number retrieved, i.e. the number of objects in $R$ of the same class as $Q$ divided by $t$. Recall refers to the ratio of the relevant objects retrieved to the total number of relevant objects in the database, i.e. the number of objects in $R$ of the same class as $Q$ divided by the total number of objects the same class as $Q$, that could be returned. In our case, the latter one is equal to the number of objects of the class of $Q$ in the database minus 1 (the query itself is not a possible answer). 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 6, 7, and 8, and Tables 1 and 2, ECCobj does a better job than ECCborder. The percentage recall of the two methods is comparable, with slightly better results from ECCobj. With respect to the average ranks, ECCobj does clearly better with the hands, humans, octopus, pliers, spectacles, spiders, teddy, and is close or slightly worse with the ants, crabs, and snakes. Both, ECCobj and ECCborder, 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 humans and octopus (see Tables 1 and 2).

Figure 8 shows comparative precision-recall results of ECCobj and ECCborder and three other methods. Again ECCobj and ECCborder are comparable, except for the teddy bears, where ECCobj is clearly superior to ECCborder. With respect to the other methods, both eccentricity based methods score better on the pliers, spectacles and snakes, while with the octopus, two of the three other methods have considerably better results.

### 4.1. Discussion

The histogram of the eccentricity transform characterizes the compactness of the shape (e.g., a flat histogram characterizes a very elongated object, a histogram with monotonically decreasing values characterizes a rather compact object - see Figure 4). This measure is used to compute similarities between shapes, robust with respect to scaling, rotation, and part articulation. The matching results are promising, especially when considering the straightforward approach.

---

**Table 1.** Average matching results using ECCobj multiplied by 100 (smaller means more similar). For each row, the first and second smallest value are printed in bold.

<table>
<thead>
<tr>
<th></th>
<th>ants</th>
<th>crabs</th>
<th>hands</th>
<th>humans</th>
<th>octopus</th>
<th>pliers</th>
<th>snakes</th>
<th>spectacles</th>
<th>spiders</th>
<th>teddy</th>
</tr>
</thead>
<tbody>
<tr>
<td>ants</td>
<td>1.75</td>
<td>5.33</td>
<td>3.72</td>
<td>3.53</td>
<td>7.20</td>
<td>2.95</td>
<td><strong>2.91</strong></td>
<td>7.25</td>
<td>5.43</td>
<td>3.72</td>
</tr>
<tr>
<td>crabs</td>
<td>5.33</td>
<td>1.55</td>
<td>3.73</td>
<td>3.50</td>
<td>3.67</td>
<td>3.02</td>
<td>4.36</td>
<td>3.99</td>
<td>3.43</td>
<td><strong>2.53</strong></td>
</tr>
<tr>
<td>hands</td>
<td>3.72</td>
<td>3.73</td>
<td><strong>2.30</strong></td>
<td>3.04</td>
<td>5.60</td>
<td>2.71</td>
<td>3.76</td>
<td>6.19</td>
<td>4.53</td>
<td><strong>2.51</strong></td>
</tr>
<tr>
<td>humans</td>
<td>3.53</td>
<td>3.50</td>
<td>3.04</td>
<td><strong>2.19</strong></td>
<td>5.03</td>
<td>2.13</td>
<td>3.15</td>
<td>5.04</td>
<td>3.52</td>
<td>2.62</td>
</tr>
<tr>
<td>octopus</td>
<td>7.20</td>
<td><strong>3.67</strong></td>
<td>5.60</td>
<td>5.03</td>
<td>3.90</td>
<td>5.02</td>
<td>6.22</td>
<td>4.04</td>
<td>4.96</td>
<td>4.58</td>
</tr>
<tr>
<td>pliers</td>
<td>2.95</td>
<td>3.02</td>
<td>2.71</td>
<td>2.13</td>
<td>5.02</td>
<td><strong>0.55</strong></td>
<td><strong>1.82</strong></td>
<td>4.97</td>
<td>3.64</td>
<td>2.11</td>
</tr>
<tr>
<td>snakes</td>
<td>2.91</td>
<td>4.36</td>
<td>3.76</td>
<td>3.15</td>
<td>6.22</td>
<td><strong>1.82</strong></td>
<td><strong>0.80</strong></td>
<td>5.73</td>
<td>4.83</td>
<td>3.55</td>
</tr>
<tr>
<td>spectacles</td>
<td>7.25</td>
<td>3.99</td>
<td>6.19</td>
<td>5.04</td>
<td>4.04</td>
<td>4.97</td>
<td>5.73</td>
<td><strong>2.24</strong></td>
<td><strong>3.97</strong></td>
<td>5.13</td>
</tr>
<tr>
<td>spiders</td>
<td>5.43</td>
<td><strong>3.43</strong></td>
<td>4.53</td>
<td>3.52</td>
<td>4.06</td>
<td>3.64</td>
<td>4.83</td>
<td>3.97</td>
<td><strong>2.25</strong></td>
<td>3.50</td>
</tr>
<tr>
<td>teddy</td>
<td>3.72</td>
<td>2.53</td>
<td>2.51</td>
<td>2.62</td>
<td>4.58</td>
<td><strong>2.11</strong></td>
<td>3.55</td>
<td>5.13</td>
<td>3.50</td>
<td><strong>1.46</strong></td>
</tr>
</tbody>
</table>

**Table 2.** Average matching results using ECCborder multiplied by 100 (smaller means more similar). For each row, the first and second smallest value are printed in bold.

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

---

![Figure 9. Examples of similar eccentricity histograms corresponding to objects of different classes. From left: octopus and spectacles, pliers and human.](image-url)
Figure 10. Example where histogram matching fails, as it is unable to capture the context of the match. Shown are ECC histograms of: (a) snake, (b), (c) human.

tion 3 more similar to (b) and to (c), than (b) to (c). Keeping in mind that these are distances, one can see that (b) should represent an object more similar to (c) than to (a), as both (b) and (c) have more “medium” and “short” distance points (more compact), while (a) is a less compact object. Please note that we have also considered other histogram matching methods ($\chi^2$ statistic and diffusion distance) and none of them produced overall significantly different results. A possible way to improve this would be to learn a distance metric [31] for matching eccentricity histograms - in this way the matching speed would not suffer too much.

In the case of 2D shapes the eccentricity of the border 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 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 ECCCobj produced better results than ECCborder can be related to the compactness of the shapes (e.g. teddy). In our approximation of the border, for very thin parts, most of the voxels of the part are also border voxels (e.g. snakes).

Compared to other approaches (e.g. [24]), one can identify the aspects discussed above (see Figure 8 and Tables 1 and 2). 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.

5. Conclusion

This paper presents matching of articulated 3D shapes using the eccentricity transform. The descriptors are defined as histograms of the eccentricity transform and matched using the $\ell_2$-norm. A big advantage of the presented approach is that it is simple and efficient. Experimental results on a database of articulated 3D shapes are given and compared to results of three other methods. We plan to study decomposition of articulated shapes using the eccentricity transform and extend the current approach to part based shape matching.

References


