

Retrieval of 3D polygonal objects based on multiresolution signatures

Roberto Lam and J.M.Hans du Buf

Institute for Systems and Robotics (ISR)
Vision Laboratory - University of the Algarve (ISE and FCT)
8005-139 Faro, Portugal

Abstract. In this paper we present a method for retrieving 3D polygonal objects by using two sets of multiresolution signatures. Both sets are based on the progressive elimination of object's details by iterative processing of the 3D meshes. The first set, with five parameters, is based on mesh smoothing. This mainly affects an object's surface. The second set, with three parameters, is based on difference volumes after successive mesh erosions and dilations. Characteristic feature vectors are constructed by combining the features at three mesh resolutions of each object. In addition to being invariant to mesh resolution, the feature vectors are invariant to translation, rotation and size of the objects. The method was tested on a set of 40 complex objects with mesh resolutions different from those used in constructing the feature vectors. By using all eight features, the average ranking rate obtained was 1.075: 37 objects were ranked first and only 3 objects were ranked second. Additional tests were carried out to determine the significance of individual features and all combinations. The same ranking rate of 1.075 can be obtained by using some combinations of only three features.

1 Introduction and related work

The increasing availability of 3D models due to technological developments allows us to use increasingly complex illustrations. Tridimensional digital scanners produce 3D models of real objects. CAD software can also produce 3D models, from complex pieces of machinery with lots of corners and edges to smooth sculptures. Very complex protein structures play an important role in pharmacology and related medical areas. The World Wide Web allows to incorporate 3D models in sites and home pages. As a consequence of this trend, there is a strong interest in methods for recognition and retrieval of 3D objects [1, 2]. Object recognition (matching) may be very time consuming because of all variations that may occur: different position (object origin), rotation, size and also mesh resolution. Similarity analysis does not require precise shape comparisons, global nor local. Normally, this approach is based on computing a set of features or a feature vector FV of a query object and comparing it with the FVs of all objects in a database.

The FVs can be obtained by a variety of methods, from very simple ones (bounding box, area-volume ratio, eccentricity) to very complex ones (curvature

distribution of sliced volume, spherical harmonics, 3D Fourier coefficients) [3–5]. The intrinsic nature of the objects may pose some constraints, and some methods may be more suitable, and faster, for the extraction of FVs than others. For example, methods based on spherical harmonics and 3D Fourier coefficients are not suitable for concave (non-star-shaped) objects, whereas other methods have problems with open (non-closed) objects. Some limitations can be solved by combining two or more methods. However, since many objects can yield very similar FVs by applying only one method, i.e., mathematically possibly an infinite number of objects, normally several methods are combined to achieve the best results. We mention the approach of [6], which is related to our own approach: they projected a 3D object onto 2D curvature maps. This is preceded by smoothing and simplification of the polygonal mesh, and final retrieval is based on comparing the 2D curvature maps. The theory of mathematical morphology (MM) arose in the middle of the 1960s [7, 8]. Developed for geometric analyses of shapes and textures, it became increasingly important in 2D image processing and computer vision. Despite all theoretical developments and generalization to 3D, most MM work is still being applied to 2D image processing [8]. The work done in 3D is rather scarce and mostly limited to three-dimensional surfaces. Jackway [9] developed an approach for the recognition of 3D objects in range data through the matching of local surfaces. Lee et al. [10] analyzed the composition of 3D particle aggregates by processing one hemisphere of the particles. In this paper we also apply MM to recognition of 3D polygonal objects, but in combination with another method, i.e., mesh smoothing.

The rest of this paper is organized as follows: Section 2 presents the proposed methods and Section 3 the experimental results. We conclude with a discussion in Section 4.

2 Overview of our approach

We use 40 objects of the AIM@SHAPE database [11]. Each one is represented by four different mesh resolutions. The models were downloaded in PLY format and they are 2-manifold, "watertight" (closed, without gaps and with regular meshes). Figure 1 shows some models and Table 1 lists all the objects and their



Fig. 1. Examples of models. From left to right: Elk, Mouse, DancingChildren, Dragon, Egea and RollingStage with increasing model resolutions.

mesh resolutions. The first three resolutions are used for creating the characteristic FV and the last resolution is used for testing in similarity search. In order to obtain invariance to scale (size) and translation, the models were normalized to the unitary sphere after the origin of object was moved to the center of the sphere. Rotation invariance is achieved by the fact that our FV is global to the model as proven in [12]. Invariance to mesh resolution is obtained by proper feature normalization, which is explained below. We apply two different methods which complement each other. Mesh smoothing affects the object’s area (Section 2.1) and the dilation-erosion method affects the object’s volume (Section 2.2).

Table 1. All 40 models with their mesh resolutions; the first three are used in resolution-invariant feature extraction, the last one is used in similarity search.

N	Model	Resolutions	N	Model	Resolutions
1	Amphora	6.5; 7.5; 9.5; 8.0	21	Fish	6.0; 7.5; 9.9; 8.0
2	Bimba	6.0; 8.5; 9.5; 8.0	22	FishA	6.0; 7.5; 9.9; 7.0
3	Blade	6.0; 7.5; 9.9; 8.0	23	Grayloc	6.0; 7.5; 9.9; 7.8
4	Block	5.0; 6.5; 8.0; 8.5	24	GreekSculpture	6.5; 7.0; 7.7; 8.5
5	Bunny	6.5; 7.5; 9.9; 8.0	25	Horse	6.0; 7.5; 9.9; 8.0
6	CamelA	6.0; 7.5; 9.9; 7.8	26	IsidoreHorse	6.0; 7.5; 9.9; 7.0
7	Carter	6.0; 7.5; 9.9; 7.3	27	Kitten	6.0; 7.5; 9.9; 7.3
8	Chair	6.0; 7.5; 9.9; 6.9	28	Liondog	6.0; 7.5; 9.9; 8.0
9	Cow2	6.0; 7.5; 9.9; 8.9	29	Maneki	6.0; 8.8; 9.8; 7.5
10	Cow	6.0; 6.4; 9.9; 7.1	30	Moai	6.5; 8.5; 9.5; 9.7
11	Dancer	6.0; 7.5; 9.9; 7.7	31	Mouse	6.0; 7.5; 9.9; 7.8
12	DancingChildren	6.0; 7.5; 9.9; 6.8	32	Neptune	6.0; 7.5; 9.9; 7.6
13	Dente	6.0; 7.5; 9.9; 7.0	33	Pulley	6.0; 7.5; 9.9; 7.0
14	Dilo	6.0; 8.5; 9.6; 7.7	34	Ramesses	6.0; 7.5; 9.9; 8.0
15	Dino	6.0; 8.3; 9.7; 7.7	35	Rocker	6.0; 7.5; 9.9; 7.1
16	Dragon	6.0; 8.0; 9.5; 7.7	36	RStage	6.0; 7.0; 9.0; 9.5
17	Duck	6.0; 7.5; 9.9; 6.7	37	Screwdriver	6.0; 7.5; 9.9; 7.0
18	Egea	7.4; 7.9; 9.5; 8.7	38	Squirrel	6.0; 7.5; 9.9; 7.2
19	Elk	6.0; 7.5; 9.9; 7.9	39	Torso	6.0; 7.5; 9.9; 7.7
20	Eros	6.0; 7.5; 9.9; 6.5	40	Vaselion	6.0; 7.5; 9.9; 7.5

2.1 Mesh smoothing

Mesh smoothing is usually used to reduce noise. [13] smoothed principal components for shape classification in 2D. In our work the main aim is related to iterative and adaptive (nonlinear) mesh smoothing in 3D. Smoothing in quasi-planar regions but not at sharp edges was used in [14] for reducing the number of vertices. Here we simply apply the *linear* version which will smooth the mesh at all vertices. It starts by eliminating very sharp object details, like protruding dents and bumps, and then after more iterations less details will remain. The sum of the displacements of all vertices, combined with the contraction ratio

of the surface area, generates a quadratic function that characterizes the model quite well.

If V_i , with $i = 1, N$, is the object's vertex list with associated coordinates (x_i, y_i, z_i) , the triangle list $T(V)$ can be used to determine the vertices at a distance of one, i.e., all direct neighbor vertices connected to V_i by only one triangle edge. If all neighbor vertices of V_i are $V_{i,j}$, with $j = 1, n$, the centroid of the neighborhood is obtained by $\bar{V}_i = (1/n) \sum_{j=1}^n V_{i,j}$. Each vertex V_i is moved to \bar{V}_i , with displacement $\bar{D}_i = \|V_i - \bar{V}_i\|$. Figure 2 shows a model and the influence of mesh smoothing. The total displacement is $D = \sum_{i=1}^N \bar{D}_i$. The entire procedure is repeated 10 times, because we are mainly interested in the deformation of the object at the start, when there still are many object details, and more iterations do not add useful information anymore. Hence, displacements are accumulated by $A_l = \sum_{m=1}^l D_m$ with $m = 1 \dots 10$. In order to obtain invariance to mesh size, in each iteration m the displacement D_m is corrected using

$$D_m := D_m \cdot \frac{NP_m \cdot N}{A_{10} \cdot S_m}, \quad (1)$$

with N the total number of vertices, NP_m the number of participating vertices (in non-planar regions which contributed to the displacement), S_m the surface of the object (sum of all triangles) after each smoothing step, and A_{10} the final, maximum accumulated displacement after all 10 iterations. Then the curve of each object and each mesh resolution is further normalized by the total contraction ratio $C = S_{10}/S_0$ (final surface and original surface), and the three curves (10 data points) are averaged over the three mesh resolutions. In the last step, the averaged A_l is least-squares approximated by a quadratic polynomial in order to reduce 10 parameters to 3. Figure 3 shows representative examples of curves A_l . It should be stressed that, in contrast to the second method as described below, no re-triangulation of the object's mesh after each iteration is done, i.e., the number of vertices—and triangles—remains the same.

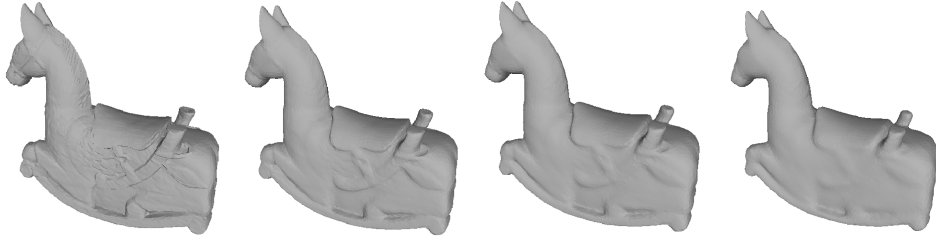


Fig. 2. Mesh smoothing applied to IsidoreHorse model. From left to right: original and smoothed meshes after 3, 6 and 10 iterations.

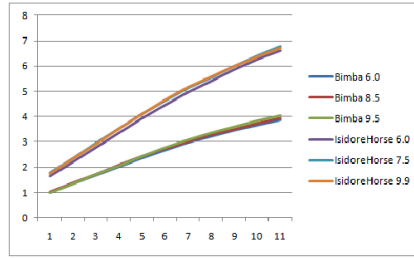


Fig. 3. Characteristic curves after mesh smoothing of the Bimba and IsidoreHorse models.

2.2 Dilation and erosion

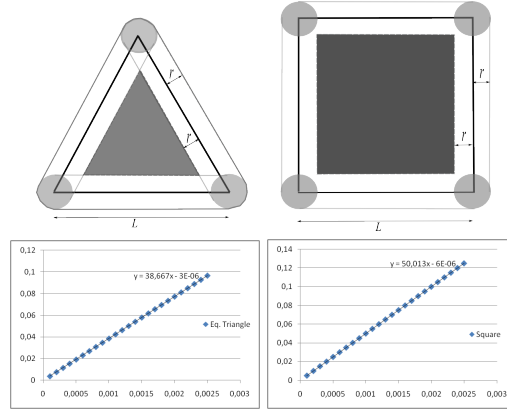


Fig. 4. Top: Erosion and dilation in 2D of equilateral triangle (left) and square (right) using a circle with radius r as structuring element. Bottom: Area β as a function of radius r of the structuring element, equilateral triangle (left) and square (right).

As in the previous section (2.1) and in [15], the basic idea of this method is to characterize 3D objects by controlled elimination of detail. This is illustrated in 2D in Figure 4. The top of the figure shows a triangle and a square with the structuring element, a circle with radius r , on the corners of the original objects. The dilated objects are bigger (only the contours are shown) and the eroded objects (shown shaded) are smaller. The surface β between both as a function of the radius r is shown in bottom: the two curves are linear but have different slopes. This effect will be exploited below in the 3D case [16].

There are a few important issues when applying mathematical morphology to 3D objects. One is associated with the type of representation: voxel or mesh [17, 18]. The voxel representation involves 3D arrays with, depending on the object's

resolution, very big dimensions, although the voxels themselves are binary: object vs. background. An advantage is that many algorithms from mathematical morphology have been developed for 2D image processing, and these can easily be adapted to 3D.

Polygonal meshes, on the other hand, have a more complex data structure. After applying the erosion and dilation operators, the new meshes must be determined, very close vertices can be collapsed, and self-intersecting facets must be detected and removed. In our method we extend boundary extraction [8] from 2D to 3D. Due to the fact that we use polygonal meshes we can apply a similar solution. If $A^c = 1 \setminus A$ is the set outside A , then

$$\beta(A) = A^c \cap (A \oplus B) + A \cap (A \ominus B)^c \quad (2)$$

is the sum of the expanded and shrunken volumes relative to the original volume, i.e., the difference volume. In order to limit distortions in the transformations, we use a sphere of which the radius r is a function of edge length. To avoid inconsistencies between different mesh resolutions, we select $r = \hat{L}/20$, where \hat{L} is an object's edge length with the maximum occurrence. This can be easily determined by filling a length histogram with 50 equal bins from L_{\min} to L_{\max} of each object. Dilations are obtained by displacing all vertices a distance r (the radius) in the direction of the normal vector. Since normal vectors always point outside, this is $-r$ in the case of erosions. Both operators are applied in two successive steps. The first step is intended to obtain the volumes of the objects after an initial erosion/dilation process. Each operator is repeatedly applied until the first self-intersection occurs. In this step we do not remove any element of the mesh, vertex nor facet.

In the second step we use the dilated (biggest) and the eroded (smallest) objects, generated in the first step, as a new starting point. The operators are repeatedly applied to the corresponding object: erosion to the smallest and dilation to the biggest object. After each erosion/dilation, we search the mesh for vertices that have a neighbor vertex in their vicinity, i.e., in the sphere with radius r centered at the vertex being processed, V_p . If there is a candidate vertex, V_c , it must be connected to V_p by at most 3 edges but it may not possess a direct edge to V_p . This restriction must be satisfied in order to keep the mesh 2-manifold. The search for the vertices with the shortest path from V_p to V_c is done by using Dijkstra's algorithm. Vertices V_p and V_c are merged by removing all edges and vertices, which causes a gap in the mesh, and then by inserting a new vertex, V_f , with coordinates equal to the average of the removed vertices. In the last step V_f is connected to the vertices forming the gap; see Fig. 5.

The elimination of self-intersecting facets is also necessary in situations where the nearest vertex is out of the vicinity sphere, the structuring element. The right side of Fig. 5 shows two situations which both lead to a self-intersection. Elimination is done using the TransforMesh Library [19], without introducing any additional deformation.

The application of a sphere as structuring element to all vertices yields a smaller object in case of erosion and a bigger one in case of dilation. The Horse

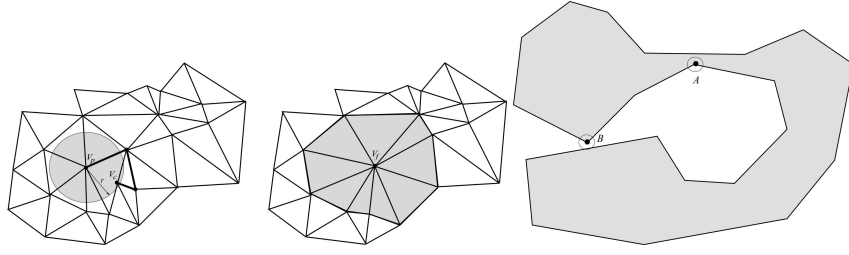


Fig. 5. Merging neighboring vertices: before (left) and after (center). The triangles around vertex A will self-intersect during erosions, and those around B during dilations (right).

model, for example, after repeated erosions will have discontinuity of the legs; see Fig. 6. The small stumps and their volumes are excluded from the computation of the Horse's parameters. The same procedure is applied to the other models. According to Eq. 2, the difference volume is defined as dilated volume minus

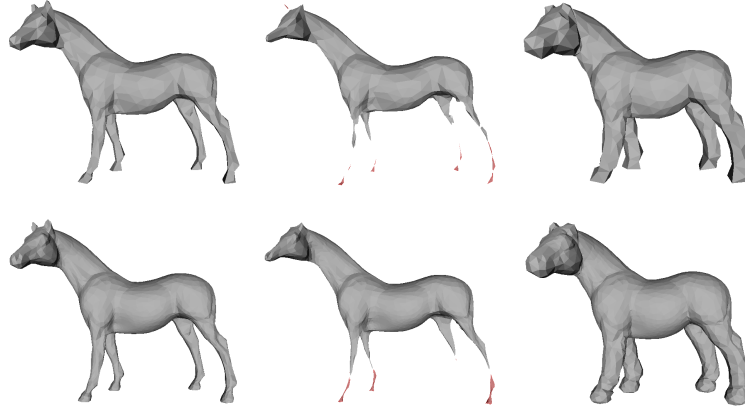


Fig. 6. Horse model: original (left), after erosion (center) and dilation (right). Mesh resolutions of 6.0 (top) and 7.5 (bottom).

eroded volume, and this yields a linear function of the radius of the structuring element; see Fig. 7. After least-squares fitting by $b_0 + b_1 r$, the slope coefficient b_1 reflects the complexity of the surface of the object. The coefficient b_0 also reflects the complexity, but with emphasis on the capacity of the object to be eroded and dilated without self-intersections, i.e., the first step of the two-step process as described above.

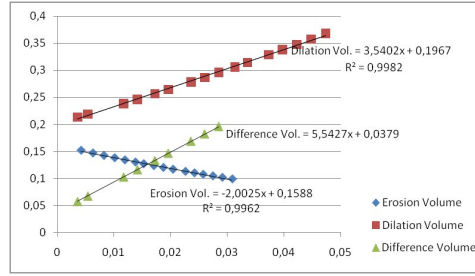


Fig. 7. Dilation-erosion function of Horse model (resolution 7.5) as function of radius.

2.3 Characteristic signatures

The 40 models listed in Table 1 are used, each with four mesh resolutions. As explained before, the first three mesh resolutions are used for constructing the FV of a model, and the last one is used for testing. Each model is characterized by 8 parameters, 5 from the method described in Section 2.1 (surface A of original model after normalization to unit sphere; contraction ratio C after 10 iterations; 3 coefficients, a_0 , a_1 and a_2 of the quadratic approximation of the smoothing curves); and 3 from Section 2.2 (volume V of original model after normalization to unit sphere; linear coefficients b_0 and b_1 of the approximated difference volume between dilated-eroded surfaces after 10 iterations). The ten iterations used in both methods were defined in order to keep the representative functions of the models well fitting to the models. Figure 8 shows typical mesh-smoothing and dilation-erosion functions.

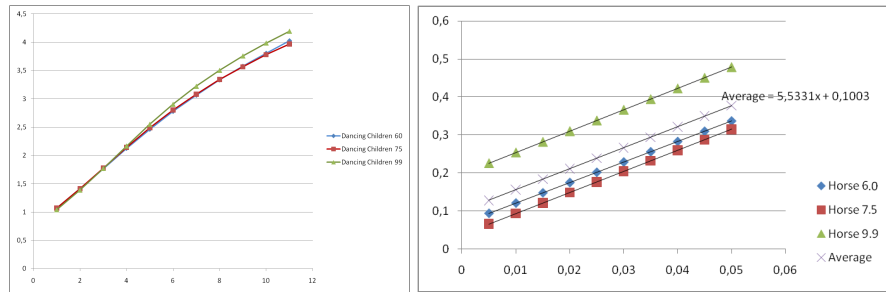


Fig. 8. Characteristic functions: mesh-smoothing function of the DancingChildren model (left) and dilation-erosion function of the Horse model.

Table 2. Ranked objects using all eight parameters. Only three objects (6, 9 and 36) were ranked second.

N	Model	Resolutions	N	Model	Resolutions
1	Amphora	1 -31-16-29-2	21	Fish	21 -10-22-3-34
2	Bimba	2 -13-30-27-29	22	FishA	22 -10-39-21-3
3	Blade	3 -22-26-21-10	23	Grayloc	23 -7-36-33-4
4	Block	4 -18-17-28-36	24	GreekSculpture	24 -25-8-10-9
5	Bunny	5 -27-13-30-1	25	Horse	25 -6-24-8-9
6	CamelA	25- 6 -24-8-15	26	IsidoreHorse	26 -3-22-21-10
7	Carter	7 -23-36-33-4	27	Kitten	27 -5-30-13-2
8	Chair	8 -25-6-24-9	28	Liondog	28 -18-17-4-40
9	Cow2	39- 9 -22-10-3	29	Maneki	29 -13-2-27-5
10	Cow	10 -21-9-39-22	30	Moai	30 -27-2-13-5
11	Dancer	11 -14-32-15-37	31	Mouse	31 -38-19-16-1
12	DancingChildren	12 -19-20-29-31	32	Neptune	32 -37-15-14-6
13	Dente	13 -27-5-30-2	33	Pulley	33 -23-7-36-4
14	Dilo	14 -15-37-11-32	34	Ramesses	34 -21-10-22-24
15	Dino	15 -37-6-32-25	35	Rocker	35 -30-27-26-5
16	Dragon	16 -38-31-19-1	36	RStage	7- 36 -23-33-4
17	Duck	17 -28-18-40-4	37	Screwdriver	37 -15-32-6-25
18	Egea	18 -17-28-4-40	38	Squirrel	38 -19-31-40-16
19	Elk	19 -12-38-31-40	39	Torso	39 -9-10-22-21
20	Eros	20 -12-29-5-15	40	Vaselion	40 -38-19-12-31

3 Results

The FVs of the objects' test resolutions were compared with the FVs of the database which were constructed by combining the three training resolutions. The objects were ranked by using the Euclidean distance between the FVs. Table 2 lists the results, starting with the object with the smallest distance, then the object with the next smallest distance, and so forth, until the fifth object. The average ranking rate $R = (1/40) \sum_{i=1}^{40} P_i$, where P_i is the ranked position of object i , is 1.075. This means that the majority of objects is ranked at position 1 or 2, at least at the first positions. Indeed, Table 2 shows that 37 objects were ranked first and only 3 second, i.e., when all eight parameters are used.

Concerning the objects ranked second, CamelA (6) was ranked after Horse (25), and RStage (36) was ranked after Carter (7). These are rather similar objects, i.e., animals and mechanic pieces, but Horse and Carter were correctly ranked first. On the other hand, Cow2 (9) was ranked after Torso (39), but these are quite different objects, and Torso was correctly ranked first.

We performed a few additional tests in order to study the significance of individual parameters and possible parameter combinations. Table 3 shows the average ranking rates of all 40 objects when each parameter is used individually. The best parameters are V (ranking rate of 1.75), b_1 (1.8), A (2.0), a_1 (2.5) and

b_0 (3.0). The discriminative power of the other three parameters is much poorer. We then did a sequential test. We took the best individual parameter V , and combined it with each of the other seven parameters. Using the best average ranking result, the best couple of parameters was selected and then combined with each of the remaining six parameters, and so on. This is not a full parameter search with all possible combinations, but it gives an impression of the most discriminative parameters. Table 4 lists the first five results. Using more than three parameters does not improve performance, i.e., there are always three objects ranked second. On the basis of Table 3 one might expect that the couple $[V, b_1]$ would be best, but Table 4 shows that the couple $[V, A]$ performs better. However, the triplet $[V, A, b_1]$ includes the best three from Table 3. Similarly, the best quadruplet $[V, A, b_1, a_1]$ includes the best four and the quintuple $[V, A, b_1, a_1, b_0]$ the best five. The remaining parameters did not improve performance, but the set of only 40 objects may be too small to draw final conclusions, apart from the fact that the best result obtained with all eight parameters is equal to that obtained with only three parameters.

Finally, in order to further validate our approach we also tested two deformed

Table 3. Average ranking rates using individual parameters.

Smoothing					Morphology		
A	C	a_0	a_1	a_2	V	b_0	b_1
2.0	11.7	6.4	2.5	8.9	1.75	3.0	1.8

Table 4. Average ranking rates obtained by a sequential combination of parameters; see text.

Parameters	Ranking rate
$[V]$	1.75
$[V, A]$	1.2
$[V, A, b_1]$	1.075
$[V, A, b_1, a_1]$	1.075
$[V, A, b_1, a_1, b_0]$	1.075

objects; see Fig. 9. Object Bimba was deformed by applying the algorithm fBM (fractal Brownian Motion, from the Meshlab package [20]) to all its vertices. Object Bunny-iH exhibits the characters i and H on its left flank; Bunny-iH is part of the AIM@SHAPE database. Both objects were correctly matched (ranked first) with the original objects.

4 Conclusions and discussion

The tested signatures—at least three of them—appear to be robust due to their global nature. In addition, small and local deformations of the object’s meshes

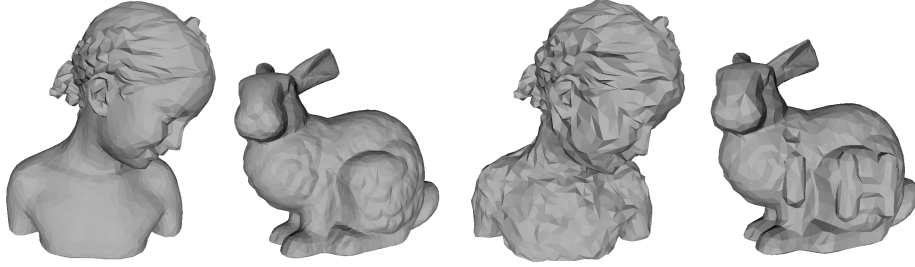


Fig. 9. Original models (left) and deformed ones (right), Bimba and Bunny-iH.

do not introduce significant modifications of the characteristic signatures, although more types of deformations must be tested with more than two objects. In general, the dataset of 40 objects tested here is too small to compute advanced performance measures as used in the SHREC contest. However, our correct recognition rate of $37/40 = 0.925$ is better than the range between 0.45 and 0.70 as achieved in the SHREC contest of 2010 [21]. Therefore, in future work the number of objects in our database should be increased such that the significance of individual parameters and the best combinations of these can be validated. In parallel, the method should be tested by using other types of objects, such as 3D meshes of complex proteins. A practical problem is that some objects are not available with different mesh resolutions, while others are not 2-manifold or "watertight" and these must be pre-processed. Another problem is that the elimination of disconnected parts after erosions (Fig. 6), which has been done manually here using Meshlab, must be automated. The latter problem does not only occur in case of e.g. animals with legs, but can be expected in case of protein structures.

Acknowledgements

This work was supported by project the FCT (ISR/IST plurianual funding) through the PIDDAC Program funds.

References

1. Bustos, B., Keim, D.A., Saupe, D., Schreck, T., Vranic, D.: Feature-based similarity search in 3D object databases. *ACM Computing Surveys* **37** (2005) 345–387
2. Tangelder, J.W., Velkamp, R.C.: A survey of content based 3D shape retrieval methods. *Multimedia Tools Appl.* **39** (2008) 441–471
3. Saupe, D., Vranic, D.V.: 3D model retrieval with spherical harmonics and moments. *Proc. 23rd DAGM-Symposium on Pattern Recognition*, London, UK, Springer-Verlag (2001) 392–397
4. Pang, M., Dai, W., Wu, G., Zhang, F.: On volume distribution features based 3D model retrieval. *Advances in Artificial Reality and Tele-Existence. LNCS 4282*, Springer Berlin / Heidelberg (2006) 928–937

5. Sijbers, J., Dyck, D.V.: Efficient algorithm for the computation of 3D Fourier descriptors. *Proc. Int. Symp. on 3D Data Processing Visualization and Transmission* (2002) 640
6. Assfalg, J., Bimbo, A.D., Pala, P.: Content-based retrieval of 3D models through curvature maps: a CBR approach exploiting media conversion. *Multimedia Tools and Applications* **31** (2006) 29–50.
7. Matheron, G.: Random sets and integral geometry. John Wiley & Sons, New York (1975)
8. Serra, J.: Introduction to mathematical morphology. *Comput. Vision, Graphics and Image Processing* **35** (1986) 283–305
9. Jackway, P.T.: Morphological Scale-Space with Application to Three-Dimensional Object Recognition. PhD thesis, Queensland University of Technology (Australia) (1995)
10. J. Lee, M. Smith, L., Midha, P.: A mathematical morphology approach to image based 3D particle shape analysis. *Machine Vision and Applications* **16** (2005) 282–288
11. AIM@SHAPE: <http://www.aimatshape.net> (2008)
12. Vranic, D.: 3D Model Retrieval. PhD thesis, University of Leipzig (2004)
13. Glendinning, R.H., Herbert, R.A.: Shape classification using smooth principal components. *Pattern Recognition Letters Volume 24*(**12**) (August 2003) 2021–2030
14. Lam, R., Loke, R., du Buf, H.: Smoothing and reduction of triangle meshes. *Proc. 10th Portuguese Computer Graphics Meeting* (2001) 97–107
15. Lam, R., du Buf, J.M.H.: Invariant categorisation of polygonal objects using multi-resolution signatures. In: *Proc. KDIR*. (2009) 168–173
16. Lam, R., du Buf, J.M.H.: Using mathematical morphology for similarity search of 3D objects. In: *Proc. of Fifth Iberian Conference on Pattern Recognition and Image Analysis. LNCS 6669*, Springer (2011) 411–419
17. Campbell, R., Flynn, P.: A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding* **81** (2001) 166–210
18. Shih, F.: Object representation and recognition using mathematical morphology model. *Journal of Systems Integration* **1** (1991) 235–256
19. Zaharescu, A., Boyer, E., Horaud, R.: Transformesh: a topology-adaptive mesh-based approach to surface evolution. In: *Proc. of Eighth Asian Conf. on Computer Vision. LNCS 4844 (2)*, Springer (2007) 166–175
20. Cignoni, P., Corsini, M., Ranzuglia, G.: Meshlab: an open-source 3D mesh processing system. *ERCIM News* (2008) 45–46
21. Veltkamp, R.C., Giezeman, G.J., Bast, H., Baumbach, T., Furuya, T., Giesen, J., Godil, A., Lian, Z., Ohbuchi, R., Saleem, W.: SHREC'10 track: Large scale retrieval. In: *Proc. of the Eurographics/ACM SIGGRAPH Symp. on 3D Object Retrieval*. (2010) 63–69