

# Surface Signature-Based Method for Modeling and Recognizing Free-Form Objects

H.B. Darbandi<sup>1</sup>, M.R. Ito<sup>1</sup>, and J. Little<sup>2</sup>

<sup>1</sup> Electrical and Computer Engineering, University of British Columbia

<sup>2</sup> Computer Science, University of British Columbia  
hosseinb@ece.ubc.ca, mito@ece.ubc.ca,  
little@cs.ubc.ca

**Abstract.** In this paper we propose a new technique for modeling three-dimensional rigid objects by encoding the fluctuation of the surface and the variation of its normal around an oriented surface point, as the surface expands. The surface of the object is encoded into three vectors as the surface signature on each point, and then the collection of signatures is used to model and match the object. The signatures encode the curvature, symmetry, and convexity of the surface around an oriented point. This modeling technique is robust to scale, orientation, sampling resolution, noise, occlusion, and cluttering.

## 1 Introduction

Object recognition can be divided into object representation and feature-matching procedures [1]. Three-dimensional model-based object representations use the geometry of the object as the basis for modeling. In this method, an object's geometric properties and relations are extracted and stored as models of that particular object. During the matching process, the same procedure is applied to the test object, and its geometric properties and relations are compared against the models for identification purposes. The main goal of all modeling techniques is to extract sufficient object features to enable reliable object recognition during the matching process.

In their early work, Besl and Jain [2] characterized the surface of an object based on the mean and Gaussian curvatures of the object's surface. Faugeras and Hebert [3] used curvature to detect primitive features in range data. The importance of curvatures in computer vision applications is well-known; however, curvatures are very sensitive to noise. Although parametric surfaces such as B-Spline [4] are very flexible, different parameters and different knots (control points) can create the same surface. Parametric surfaces are more suitable for computer graphics. All EGI-based techniques such as [5] are generally considered global techniques, and they react poorly in the event of occlusion in the recognition process. These modeling techniques are not unique, except for CEGI [6]. Point signatures [7] are vulnerable to surface sampling and noise, and can therefore result in ambiguous representations [8]. The splash representation [9] is also sensitive to noise and occlusions. The symmetrical properties of Super Quadrics [10] limit their capability in modeling free-form objects. Constructive Solid Geometries [11] risk the ambiguity in joining volumetric primitives.

Recently, due to the decreasing cost of 3D scanners, more complex methods that use patches created from dense images have been introduced in the literature. These methods include spin images [12], surface point signatures [13], harmonic shape contexts [14], and the tensor method, which models and recognizes 3D objects in cluttered environments [15]. A good survey of different techniques can be found in [16].

Of the more recently introduced methods, the spin image is perhaps the most accurate and easy to implement. The spin image reacts very well in the event of occlusion and cluttering, and is also robust to noise. Spin image representation is sensitive to the resolution and sampling of the models and scene. An improvement to this technique [17] overcomes the sensitivity of the spin image to resolution and sampling. Spin images map a 3D surface into a 2D histogram. Histograms do not encode surface geometric properties, which are essential in many computer vision applications. Furthermore, 2D histograms lead to large model size and an inefficient matching process.

The technique proposed in this paper can be used to model a free-form object. This modeling technique is not only robust to scale, orientation, occlusion, cluttering, sampling resolution, and noise, but the resulting models are also packed and much smaller than those created by spin images. Furthermore, the proposed modeling technique encodes the curvature, symmetry, and convexity of the surface around an oriented point that can be used in other related applications. The proposed technique encodes the surface into three discriminating vectors, making the matching process more efficient and accurate.

The rest of this paper is organized as follows. In Section 2, we describe the proposed modeling technique, parameters, and data set. In Section 3 we present the results and analysis of the experiments.

## 2 Flatness and Orientation Signatures

The basic element used to model and recognize an object in this paper is an oriented point. The oriented point is a point ( $P$ ) on the surface of an object along with its normal ( $N$ ) at point  $P$ .

Consider an oriented point,  $P$ , on the surface of an object (please refer to the left-hand column in Figure 1). Now, assume a sphere with radius  $R$ , centered on  $P$ . If  $S$  is the total area of the object circumscribed inside the sphere, and  $A$  is the projection of  $S$  on a plane  $\Pi$ , normal to  $N$ , then the length of the normal,  $N$ , is set to

$$\|N\| = F = \frac{A}{S} \quad F \leq 1 \quad (1)$$

$F$  specifies the flatness of the area around an oriented point,  $P$ , on the surface. For a flat surface,  $F$  is equal to 1; for a curved surface,  $F$  is less than one. The more curved the surface, the lower the value of  $F$ .

As  $R$  increases, the ratio of  $A/S$  changes, creating a graph that is the flatness signature of the surface around point  $P$ . The flatness signature of a flat area is a horizontal line. Similarly, the angle of the normal,  $\theta$ , which is equal to the average of the normal of the patches enclosed in the sphere, also fluctuates from its original position,  $N$ , as  $R$  increases, creating another curve, the orientation signature. For a

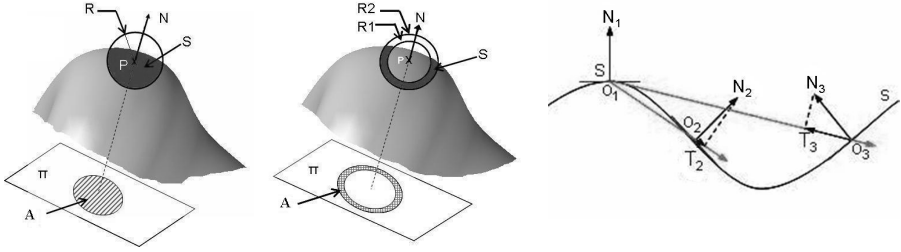


Fig. 1. Modeling Technique

symmetric surface, the orientation signature is a horizontal line if P is set on the symmetrical point on the surface. The combination of orientation signature and flatness signature models the object on point P on the surface of an object. The collection of the signatures is used to model the entire object. Each signature is a vector of  $n$  elements, in which  $n$  is the number of intervals used to generate the signature. For each interval, the radius of the sphere is set to  $R_i = R_{i-1} + \Delta R$ .

To find the signature for each oriented point on a vertex, the normal of the vertex,  $N$ , is first calculated by averaging the normal of the surfaces around the vertex. To decrease noise effect, the normal of the patches is averaged inside a sphere with a specific radius called the *base radius*. By finding the normal,  $N$ , and  $\Pi$ , the plane normal to  $N$ , the two vectors  $S = [s_i \geq 0 \mid s_1, \dots, s_n]$  and  $A = [a_i \geq 0 \mid a_1, \dots, a_n]$  can be calculated. The flatness signature,  $F = [f_i \geq 0 \mid f_1, \dots, f_n]$ , of the oriented point P is then calculated from  $F = A / (S + \epsilon)$ . Here,  $\epsilon$  is added to each element of  $S$  to avoid division by zero.

Concurrently, the oriented signature  $O = [o_i \geq 0 \mid o_1, \dots, o_n]$  is found. The collection of  $F$  and  $O$  signatures models the surface of the object at the selected vertex.

The main problem with this method is that as  $R$  increases, the signatures reach a relatively steady state. Consequently, the discriminativity of the signatures decreases. To overcome this problem, we use the surface of the object circumscribed between two steps of  $R$  for signature creation, rather than accumulating the surface as  $R$  increases. Instead of a single sphere, two co-centered spheres are used, as shown in the middle column of Figure 1. The area circumscribed between two spheres within radius  $R_2 - R_1$  is then used for signature creation.

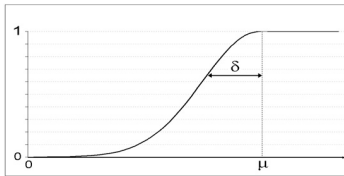


Fig. 2. Filter used to smooth flatness signatures

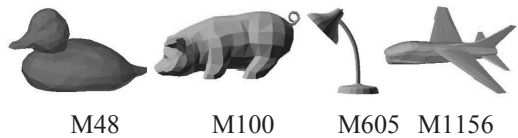


Fig. 3. Test models

Experiments show that the signatures created by this method are sensitive to noise. When  $s_i$  and  $\alpha_i$  are too small, a small fluctuation on the surface of the object caused by noise affects the flatness signature. To remedy this problem, the flatness signatures are smoothed with a Gaussian shape filter (Figure 2) with parameters  $\mu = \text{mean}(S)$  and  $\delta = \text{std}(S)$  for all  $s_i < \mu$ .

## 2.1 Convex and Concave Surfaces

The method we have introduced so far creates identical signatures for both concave and convex surfaces. To distinguish between concave and convex surfaces, the surface is divided into positive and negative patches (convex and concave surfaces respectively) based on the direction of their normal relative to an oriented point. Consider the cross-section of surface  $S$  in the right-hand column of Figure 1.  $O_1$  is a point on the surface, and  $N_1$  is its normal. Let us assume we are interested in finding the signatures of the surface relative to oriented point  $O_1$ . To find the convexity and concavity of the surface on points  $O_2$  and  $O_3$  relative to  $O_1$ , connect  $O_1$  to  $O_2$  and  $O_1$  to  $O_3$  to create two vectors,  $O_1O_2$  and  $O_1O_3$ . Then find the projection of  $N_2$  and  $N_3$  on  $O_1O_2$  and  $O_1O_3$ ,  $T_1$  and  $T_3$ , respectively. If the direction of  $O_1O_x$  and  $T_x$  are the same, then the surface at point  $O_x$  is convex relative to point  $O_1$ . If the direction of  $O_1O_x$  and  $T_x$  are opposite, then the surface at point  $O_x$  is concave relative to  $O_1$ . For example, the surface at  $O_2$  and  $O_3$  are convex and concave, respectively, relative to the oriented point  $O_1$ . In addition to flatness and orientation signatures, the total convex and concave area at each step of signature creation is stored separately. The calculation of convex and concave areas has no major effect on processing time since the same procedures are used for calculating flatness and orientation signatures.

The convexity signature is calculated by dividing the convex area to the total area at each step of signature creation. The convexity can hold a value between one and zero. A value of one for convexity means that the area being considered for signature creation is a complete convex area. A value of zero means that the area is a complete concave or a flat area. Convexity signatures, along with flatness and orientation signatures, are used to model the surface around an oriented point.

## 2.2 Parameters of the Model

The signatures created with the proposed modeling technique depend on two parameters. Support distance and support angle [18] are two parameters that limit the modeling area of a surface around an oriented point. The support distance limits the value of  $R$ . Small values of  $R$  model the local deformation around point  $P$ , and large values of  $R$  model the global deformation of the surface relative to the oriented point  $P$ .

Because of occlusion, we cannot see the entire object from a single viewpoint. It is logical to assume that if the normal of a patch makes an angle greater than a threshold angle with the orientation of point  $P$ , it cannot be seen from the same angle that sees point  $P$ . As a result, those patches cannot be used for modeling the object from that viewpoint. This parameter is called the support angle.

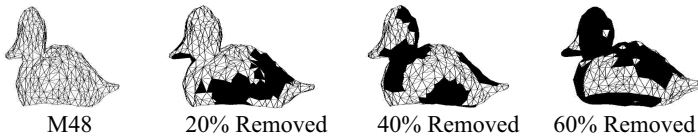
### 2.3 Models and Settings

We chose M48 [19], M100, M605, and M1153 (a duck, a pig, a stand light, and a toy airplane) as our experimental models (Figure 3). The dimensions of these models are listed in Table 1. Normal noise with standard deviation equal to 4 (2% of the maximum dimension of the model) was added to each vertex in the normal direction of the surface of M48, M100, M605, and M1153 to create M48N4, M100N4, M605N4, and M1153N4 respectively.

**Table 1.** Dimensions of test models

Model	Length	Height	Width	Number of Patches	Number of Vertices
M48	97.5	90.7	200	1307	671
M100	112.2	91.1	200	1208	606
M605	100.4	200	170.9	1300	531
M1153	161.3	71.5	200	1478	743

To demonstrate the signatures' robustness to clutter, a large portion of the test object was removed by selecting one to five random points as seed patches on the surface of the test object; surface patches were removed of up to 60% of the total surface of the object (Figure 4).



**Fig. 4.** M48 with 20%, 40%, and 60% of the surface patches removed

### 2.4 Similarity Measurement

To compare the signatures of the models, we multiplied the angle of the normalized cross-correlation and the normalized Euclidian distance of the signatures:

$$similarity = \cos^{-1}(NCC(S_A, S_B)) \times \frac{|S_A - S_B|}{length(S)} \quad (2)$$

To compare the oriented points, we used the Euclidian distance between the similarity measurements:

$$likeness = \|O, \alpha F, \beta C\| \quad (3)$$

where  $F$ ,  $O$ , and  $C$  are the *similarity* measurements for the orientation, flatness, and convexity signatures, respectively. The coefficients  $\alpha$  and  $\beta$  are weights of the similarity measurements, and their values depend on noise level. In our experiments both coefficients were set to one.

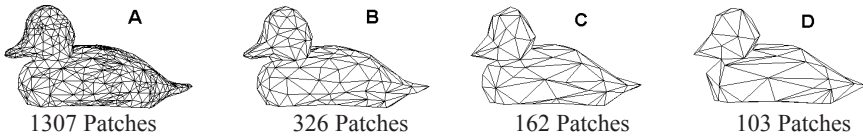
### 2.5 Patch Resolution

The size of the patches has no significant effect on the creation of the signatures. To illustrate the effect of patch size on surface signatures, M48 was sampled with 1307, 326, 162, and 103 patches, labeled as A, B, C, and D, respectively, as shown in Figure 5.

**Table 2.** Results of *likeness* measurement of signatures M48 (A) with the signatures of models M48N4, B, C, and D.  $\mu$  and  $\delta$  stand for mean and standard deviation of the comparisons, and the results are shown in  $\log_{10}$

		M48N4	B	C	D
likeness	$\mu$	-1.55	-2.09	-1.71	-1.54
	$\delta$	-1.49	-1.91	-1.62	-1.50

The results of *likeness* comparisons of model A with corresponding points from models B, C, D, and M48N4 are summarized in Table 2. The results are shown in  $\log_{10}$ . As indicated in the table, compared with model A the effect of sampling resolution on the signature creation of the models labeled B, C, and D is less than or equal to the effect of the noise on signature creation added to M48N4. As will be shown in Section 3, the proposed modeling technique is robust to this level of noise.

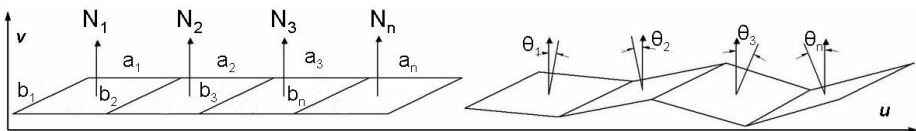


**Fig. 5.** M48 sampled with different sampling rates

### 2.6 Noise Effect

Because of averaging in our modeling technique, noise was suppressed. In Figure 6 the left-hand column shows samples of patches with their normal, and the right-hand column shows the same patches perturbed by noise. The oriented signature of an oriented point in each interval of signature creation is calculated as follows:

$$O_i = \frac{1}{m} \sum_{j=1}^m s_j N_j = \frac{1}{m} \sum_{j=1}^m a_j b_j \begin{bmatrix} u_j \\ v_j \end{bmatrix} \tag{4}$$



**Fig. 6.** Left: patches along with their normal. Right: the same patches fluctuated by noise

where  $s_j = a_j b_j$  is the surface of the patch,  $N_j$  is its normal,  $m$  is number of the patches, and  $u_j$  and  $v_j$  are components of  $N_j$  in  $u$  and  $v$  directions.

For the purposes of this manuscript calculations were simplified by making an assumption that noise affects two adjacent vertices in only the  $v$  direction with the same amount of noise,  $g_j$ , as shown in the right-hand column of Figure 6. Then

$$s'_j = b_j \sqrt{a_j^2 + g_j^2} \text{ and}$$

$$O'_i = \frac{1}{m} \sum_{j=1}^m s'_j N'_j = \sum_{j=1}^m s'_j \begin{bmatrix} p_j & -q_j \\ q_j & p_j \end{bmatrix} N_j \quad (5)$$

where  $p_j = a_j / \sqrt{a_j^2 + g_j^2}$  and  $q_j = g_j / \sqrt{a_j^2 + g_j^2}$ .  $N_j$  is the normal of the patch before noise effect. By simplifying equation (5), then

$$O'_i = \frac{1}{m} \sum_{j=1}^m \begin{bmatrix} a_j b_j & -b_j g_j \\ b_j g_j & a_j b_j \end{bmatrix} \begin{bmatrix} u_j \\ v_j \end{bmatrix} = \frac{1}{m} \sum_{j=1}^m a_j b_j \begin{bmatrix} u_j \\ v_j \end{bmatrix} + \frac{1}{m} \sum_{j=1}^m g_j b_j \begin{bmatrix} -v_j \\ u_j \end{bmatrix}$$

Finding the difference between two oriented signatures, and using Cauchy-Schwarz inequality

$$\|e\| = \|O_i - O'_i\| = \left\| \frac{1}{m} \sum_{j=1}^m g_j b_j \begin{bmatrix} v_j \\ -u_j \end{bmatrix} \right\| \leq \frac{1}{m} \sqrt{\sum_{j=1}^m b_j^2} \sqrt{\sum_{j=1}^m g_j^2} = b \sqrt{\frac{\sum_{j=1}^m g_j^2}{m}} = b \delta \quad (6)$$

where  $\delta$  is the standard deviation of noise. Given that for simplicity it is assumed in Figure 6 that  $b_1 = b_2 = \dots = b_m = b$ .

As shown in Equation (6), by increasing the number of samplings, and consequently decreasing the size of  $b$ , the noise effect can be controlled.

We have shown the effect of noise on an orientation signature with noise applied in one dimension. However, it can be demonstrated that the findings also apply when noise is applied in all three dimensions. Furthermore, it can be shown that the effects of noise on flatness and convexity signatures are also suppressed, although flatness signatures are more sensitive to noise than orientation and convexity signatures.

### 3 Object Recognition

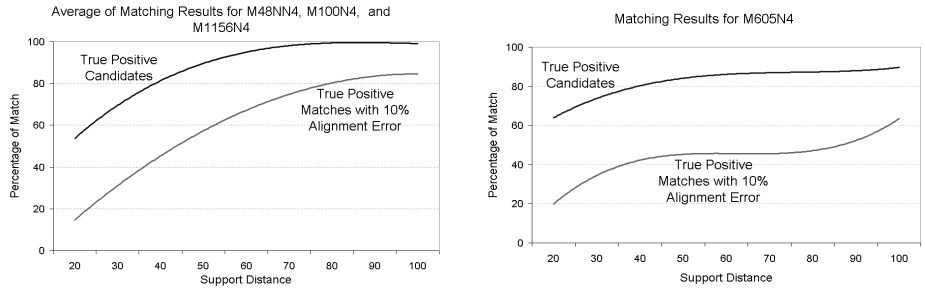
*Likeness* measurement provides a means of finding the corresponding points on the surface of a test object and models. To match the signatures, first the flatness, orientation, and convexity signatures created from a sample oriented point on the surface of the test object are compared to the flatness, orientation, and convexity signatures of all models, using Equation (2). Then their likeness is measured using Equation (3). Our experiments show that the signatures created from an object with the proposed modeling technique are more similar to the signatures created from the same object than they are to the signatures created from other objects. During the matching process, a sample point can be matched to more than one point for two reasons: both the points located on symmetrical parts of an object, and similar surfaces, create similar signatures. To counter this problem, the maximum of the top  $M$  likeness matches for each sample point within the threshold is selected, and

outliers are removed by geometric verification. A histogram of the models to which the verified matches belong is then created. The model that has the highest value in the histogram is selected as the *candidate model*. The candidate model is then passed to the registration process [20] for alignment. If the average error of alignment between the test object and the selected model falls below a threshold value, the candidate is selected as the *matched model*.

### 3.1 Experiments

In our experiments, six sample points were selected on the surface of the test object. The threshold value was set from 0.1% to 0.2% of the total signatures, and M was set to 4. Support distance was set from 20 to 100 with steps of 5, and the support angle was set to  $\pi/3$ . The average alignment error and base radius were set to 10% and 5% of the maximum dimension of the objects respectively.

Since we were considering only one object, the results were either true-positive or false-negative. The library models consist of 69 toys, with 56663 vectors for each flatness, orientation, and convexity signature. Each experiment was conducted 50 times for each support distance, and each graph in Figure 7 is the outcome of 450 to 1800 recognition tries. The curves are smoothed by a polynomial trend of order 2.



**Fig. 7.** Left: Average of matching results for M48N4, M100N4, and M1156M4. Right: Matching results for M605N4. Support angle set to  $\pi/3$ .

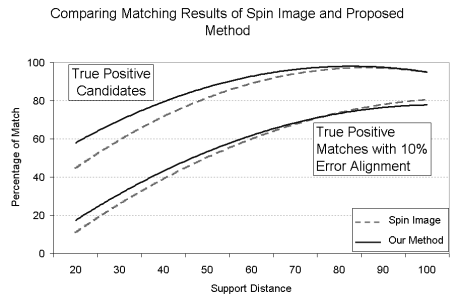
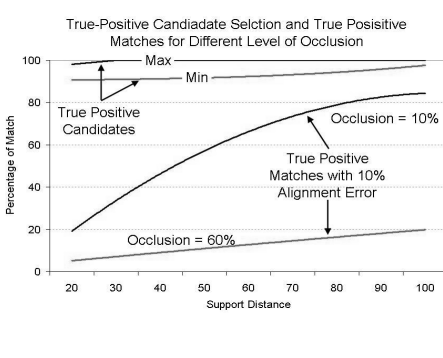
### 3.2 Results

Figure 7 shows recognition results for M48N4, M100N4, M605N4, and M1156N4. Each figure shows the result of positive candidate selection for each support distance and positive match, with an average of 10% alignment error.

The left column of Figure 7 shows the average matching results for M48N4, M100N4, and M1156N4. As indicated in the graph, the positive candidate selection begins at 50% for support distance 20 and rises to 100% as support distance increases. However, because of the threshold set to alignment error, the positive match is very low for small support distances. Experiments indicate that when the threshold of alignment error or the number of sample points selected on the surface of the test object are increased, the result improves and moves closer to positive candidate curves.

As indicated in the right column of the figure, the matching results for M605N4 are very low compared to the matching results for other test objects. This is because M605 is a symmetrical elongated object, and its signatures are very similar to each other. The signatures selected on the surface of the test object match with multiple signatures on the surface of the model, causing the alignment process to fail for the specified threshold values. This problem can be addressed by selecting extra points on the surface of the test object after positive candidate selection for a better alignment process. Figure 7 also indicates that the positive match for this model increases for support distances above 80, because the sphere reaches the boundary of the object, and the signatures are more descriptive at the boundaries for this kind of object.

Figure 8 displays the recognition results for the experiment with cluttered objects. Here we assumed that the cluttered areas had been segmented and removed, so these areas were not used for signature creation. The test object, M48N4, was cluttered randomly from 10% to 60% of its total surface, and then 6 random oriented points were selected on the uncluttered surface of the object. The signatures of the sample oriented points were matched with the signatures of the models in the library. The support angle was set to  $\pi/3$ . The figure shows the true positive match with 10% alignment error for occlusions of 10% and 60%. For greater clarity, the occlusions between 10% and 60%, which fall between the two curves, are not shown here. The figure also shows the maximum and minimum positive candidate selections for each support distance.



**Fig. 8.** Matching results for occluded test object

**Fig. 9.** Comparing matching results of spin images, and proposed modeling technique

### 3.3 Comparison

The spin images of the library models and test models were created with the same parameters. We then used the same algorithm, chose the same sample-oriented points, and repeated the recognition process with the same parameters. The average results for recognizing test objects with spin images and with the proposed modeling technique are shown in Figure 9.

As indicated in Figure 9, positive candidate selection improved by 12.5% for support distance 20 and 0% for support distance 100, and true positive matches improved by 8.5% for support distance 20, and decreased by -1.5% for support

distance 100. The spin images used in the experiments were not compressed, and the models and point selections were exactly the same for both methods.

The size of the proposed model used in our experiments is 7.5% of the size of the model used to create spin images. Spin images use Principal Component Analysis (PCA) [21] to compress modeling data, and a similar technique can be used to pack the data in the modeling method introduced in this paper.

The process of signature creation is complicated and requires considerable processing time. The time needed to model an object is ( $m^2$ ), where  $m$  is the number of the model's vertex. However, the modeling process is performed offline. In our experiment, each signature for the test model (M48N4), consisting of 671 vertexes, required an average of 6 seconds on a PC with a Centrino 1.8GHZ processor with 1GB RAM. Each matching process, which consisted of selecting 6 random points and generating their signatures, matching, and verification, required approximately 40 seconds.

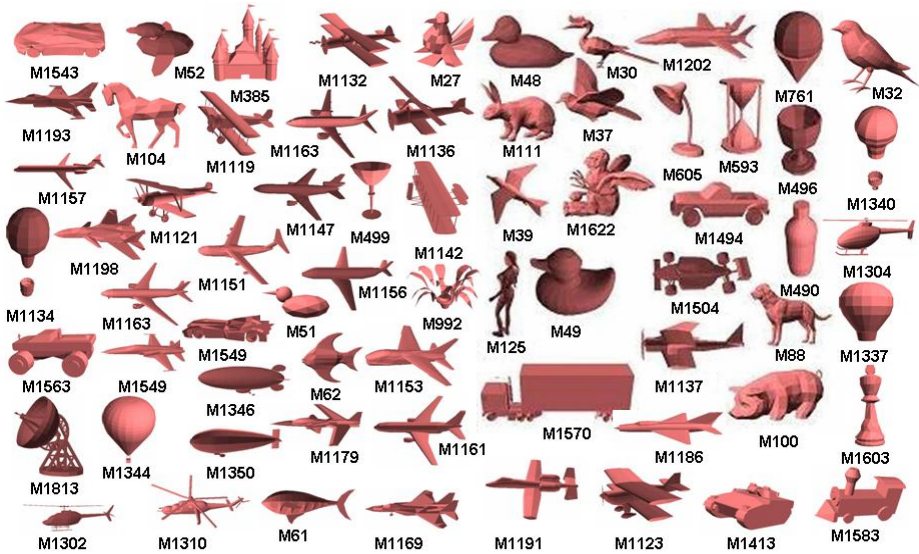


Fig. 10. Library models used in the experiments

## 4 Conclusions

In this paper we present a general method for modeling and recognizing 3D objects. The representation we describe is simple but rich enough to model and match free-form objects effectively. The signatures obtained in the experiments show that while this technique provides an elegant method for modeling and matching objects, it also encodes the curvature, symmetry, and convexity of the surface around an orientation point that can be used in related applications. The modeling and matching parameters can be adjusted to provide the results best suited to the particular experimental subject. However, the modeling process is more complicated, and uses more CPU cycles.

The results presented in the paper indicate that the matching process involves two major parameters: the number of oriented points selection, and the number of maximum top match points selection ( $M$ ). Increasing the cardinality of these two parameters improves the matching result.

The proposed method is not only applicable to object recognition: It can also be used to trace an object. Furthermore, the signatures created by the model can be used to find the symmetrical access of 3D objects and spot the high curvatures and flat areas on an object's surface. Our experiments show that the signatures can be grouped to create a pool of data that can be shared to model a variety of objects with limited amounts of data. This application needs further investigation.

## References

1. Bennamoun, M., Mamic, G.J.: *Object Recognition: Fundamentals and Case Studies*. Springer, Heidelberg (2000)
2. Besl, P.J., Jain, R.: Range Image Understanding. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 430–449 (1985)
3. Faugeras, O.D., Herbert, M.: The Representation, Recognition, and Location of 3-D Objects. *The International Journal of Robotics Research* 5(3), 27–52 (1986)
4. Wand, J., Cohen, F.S.: Part II: 3-D Object Recognition and Shape Estimation from Image Contours Using B-Splines, Shape Invariant Matching, and Neural Network. *IEEE Trans. Pattern Analysis and Machine Intelligence* 16(1), 13–23 (1994)
5. Horn, B.: Extended Gaussian Image. *Proceedings of the IEEE* 72, 1671–1686 (1984)
6. Kang, S.B., Ikeuchi, K.: The complex EGI: A new representation for 3D pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15(7), 707–721 (1993)
7. Chua, C.S., Jarvis, R.: Point Signatures: A New Representation for 3D Object Recognition. *Int'l J. Computer Vision* 25(1), 63–85 (1997)
8. Mian, A.S., Bennamoun, M., Owens, R.A.: Automatic Correspondence for 3D Modeling: An Extensive Review. *Int'l J. Shape Modeling* (2005)
9. Stein, F., Medioni, G.: Structural Indexing: Efficient 3D Object Recognition of a Set of Range Views. *IEEE transactions on pattern analysis and Machine Intelligence* 17(4), 344–359 (1995)
10. Katsoulas, D., Kosmopoulos, D.I.: Box-like Superquadric Recovery in Range Images by Fusing Region and Boundary Information. *Pattern Recognition*, 2006. In: *ICPR 2006. 18th International Conference on, August 20-24, vol. 1*, pp. 719–722 (2006)
11. Haralick, R.M., Shapiro, L.G.: *Computer and Robot Vision*. Addison-Wesley, Reading (1993)
12. Johnson, A.E.: *Spin Image: A Representation for 3D Surface Matching*. PhD Thesis, Carnegie Mellon University (1997)
13. Correa, S., Shapiro, L.: A New Signature-Based Method for Efficient 3D Object Recognition. *Proc. IEEE Conf. Computer Vision and Pattern Recognition* 1, 769–776 (2001)
14. Yamany, S.M., Farag, A.: Freeform Surface Registration Using Surface Signatures. *Proc. Int. Conf. on Computer Vision* 2, 1098–1104 (1999)
15. Bennamoun, A.S., Owens, M.: Three-Dimensional Model-Based Object Recognition and Segmentation in Cluttered Scenes Mian. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28(10), 1584–1601 (2006)

16. Campbell, R.J., Flynn, P.J.: A Survey of Free-Form Object Representation and Recognition Techniques. *Computer Vision and Understanding* 81, 166–210 (2001)
17. Carmichael, O., Huber, D., Hebert, M.: Large Data Sets and Confusing Scenes in 3-D Surface Matching and Recognition. In: *Proc. Int'l Conf. 3-D Digital Imaging and Modeling*, pp. 358–367 (1999)
18. Johnson, A., Herbert, M.: Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. *IEEE Tr. on Pattern Analysis and Machine Intelligence* 21(5) (1999)
19. <http://shape.cs.princeton.edu/benchmark>
20. Nene, S.A., Nayar, S.K.: A simple Algorithm for Nearest Neighbor Search in High Dimensions. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 999–1003 (1997)
21. Forsyth, D.A., Ponce, J.: *Computer Vision: A Modern Approach*. Prentice-Hall, Englewood Cliffs (2000)