

Shape Spectrum Based View Grouping and Matching of 3D Free-Form Objects

Chitra Dorai, *Member, IEEE*, and Anil K. Jain, *Fellow, IEEE*

Abstract—We address the problem of constructing view aspects of 3D free-form objects for efficient matching during recognition. We introduce a novel view representation based on “shape spectrum” features, and propose a general and powerful technique for organizing multiple views of objects of complex shape and geometry into compact and homogeneous clusters. Our view grouping technique obviates the need for surface segmentation and edge detection. Experiments on 6,400 synthetically generated views of 20 free-form objects and 100 real range images of 10 sculpted objects demonstrate the good performance of our shape spectrum based model view selection technique.

Index Terms—Free-form objects, sculpted surfaces, 3D object representation, COSMOS, shape spectrum, clustering, view matching.



1 INTRODUCTION

A 3D rigid object can give rise to arbitrarily many different 2D appearances (views). For an object with free-form or sculpted surfaces, only a part of a surface will typically be visible from a single viewpoint due to the object's curvedness. Variations in viewing directions and angles can result in very distinct views of the object, with more of the curved surface(s) either coming into the view or disappearing from the view. A free-form surface is defined as a smooth surface on which the surface normal is well defined and continuous almost everywhere, except at vertices, edges, and cusps [1]. The surface is not constrained to be polyhedral, piecewise-quadric, or superquadric, and its shape can be arbitrary. Some representative free-form objects are human faces, cars, boats, airplanes, and sculptures. Fig. 1 shows a set of range images of 3D free-form objects which were obtained using a laser range scanner (Technical Arts White scanner) that produces depth data in an X - Y grid. The figures show surface depth as pseudo intensity, displaying the relative orientation of the surfaces; points oriented almost vertically are shown in darker shades.

With growing interest in automated manufacturing and inspection, representation of free-form objects is gaining a lot of attention [1], [2]. Previous approaches to 3D object representation can be categorized as either *viewpoint-independent* (object-centered) or *viewpoint-dependent* (viewer-centered). A viewpoint-independent representation attaches a coordinate system to an object; all points or object features are specified with respect to this system. The description of the object thus remains canonical, independent of the viewpoint. However, it is difficult to derive an object-centered representation from an input image. A unique coordinate system needs to be first identified from the input images, and this becomes difficult when the object has many natural axes. Therefore, this approach is well suited to simple 3D objects that can be specified by analytic functions.

A viewer-centered approach, on the other hand, describes an object relative to the viewer; as one does not have to compensate for

the viewpoint, the representation can be easily computed from the image. A major disadvantage is that a large number of views needs to be stored for each object, since different views of an object are, in essence, treated as containing distinct objects. However, representing an object with multiple views is quite useful for view-based matching, alleviating the need for expensive 3D model construction.

When representing a free-form object, viewer-independent schemes suffer greater difficulty. A free-form object may neither have a complete geometric model nor a description in terms of analytic functions. It may not be an assembly of simple surfaces, such as planes and cylinders. Therefore, a practical solution would be to build a multiple view based representation of the object. However, a sculpted object gives rise to infinitely many different views owing to its curved nature. In practice, only a finite number of such views can be stored. Therefore, an important issue is which and how many of these views are actually necessary and useful for *object recognition*.

The problem we address in this paper is as follows: Given a set of views (range images) of a 3D free-form rigid object, how do we represent and organize these views in a meaningful and efficient manner? Specifically, how do we generate a *representative and adequate grouping* of the views such that a new object view can be indexed effectively and efficiently to one or a few of the stored views in the database? We place emphasis on automatically obtaining the clusters of views without requiring segmentation of object surfaces. Such a set of view clusters will serve as a view-based representation of each object in the database. Our focus is on generation of a set of representative views suitable for efficient retrieval, rather than for geometric reasoning or other purposes. Efficient retrieval of a cluster of views provides a set of plausible correct matches for further refined matching. The term *view* refers to a range image of an unoccluded object obtained from any arbitrary viewpoint. For the purposes of this paper, two views of an object are not considered distinct if they produce appearances of the object that merely differ from each other by a rotation about the view plane.

2 PREVIOUS WORK

To construct a multiple view based description of an object, an “approximate visibility technique” is adopted to restrict the set of possible viewpoints to a sphere of large but finite radius, centered around the object. The surface of the viewing sphere is tessellated in a quasiregular fashion to provide a discrete set of points which provide viewpoint vectors in the approximate visibility space. In this paper, range data of an object surface seen from each of the sampled view directions are obtained using the laser range scanner or from the CAD model or a surface triangulation of the object.

View clustering for a single object has been addressed by several researchers under the topics of characteristic views (CVs) [3] and aspect graphs (AGs) [4]. Although there exists an extensive body of work dealing with construction of aspect graphs of polyhedra and a class of curved surfaces [5], [6], [7], [8], [9], a major difficulty that confounds the use and implementation of AGs for object recognition is that complicated objects can result in enormous and complex AGs. To derive aspect graphs of manageable sizes, appropriate heuristics need to be designed. The problem of computing the aspect graph of an arbitrary object still remains unsolved. Ikeuchi's practical approach [10] relied on detecting the planar and curved faces of an object, using photometric stereo in order to form the aspects of the object containing topologically similar views. However, such an approach is difficult with a free-form object, since each viewpoint gives rise to a slightly different view of the object, because of its smoothly curved nature. It is also hard to define a *single face* in a sculpted object. A recent approach

- C. Dorai is with the IBM T.J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598. E-mail: dorai@watson.ibm.com.
- A.K. Jain is with the Department of Computer Science, Michigan State University, East Lansing, MI 48824-1027. E-mail: jain@cps.msu.edu.

Manuscript received 3 July 1996. Recommended for acceptance by K. Bowyer. For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 104917.

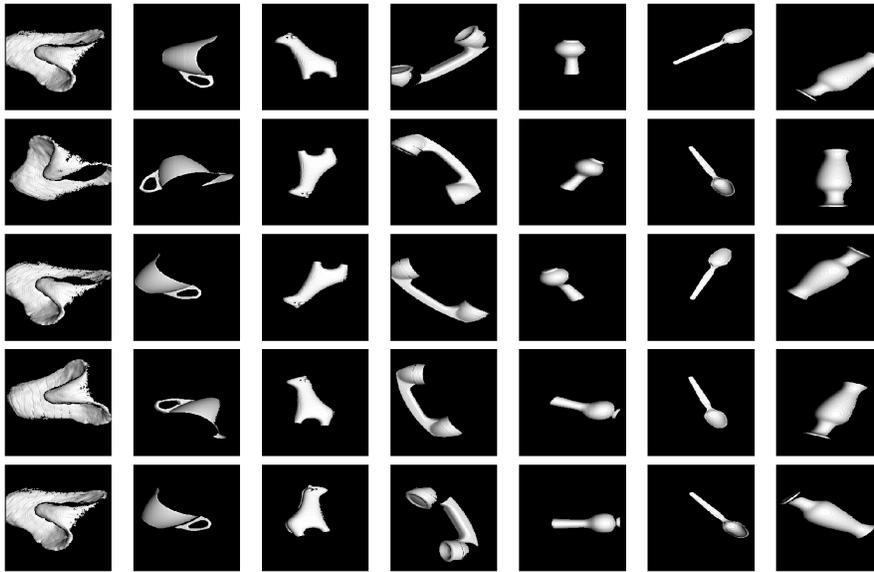


Fig. 1. Range images of views of 3D free-form objects.

[11] organizes the model base hierarchically, using parametric structural descriptions built from the CAD models of objects, where it is assumed that a complete 3D description of an object is available for its recognition. Other approaches to organizing object views for indexing include [12], [13], [14], [15], [16], but space limitations preclude a detailed discussion. In brief, our approach differs from them in terms of the features used (we use the shape spectrum for the first time), the usage of scale information (we normalize instead), construction of view aspects of volumetric parts of the object (we use shape summaries of whole object views), modelbase dependent feature prediction hierarchies and decision trees (in our case view grouping and modelbase organization are carried out independently), and object domain limitations (we report experiments on real free-form objects).

3 COSMOS AND THE SHAPE SPECTRUM

Many of the representational problems associated with free-form objects have been addressed in a scheme called COSMOS. We briefly review a few basic concepts of COSMOS, referring the reader to [2] for details. We begin with the definition of the *shape index*, a quantitative measure of the shape of a surface at a point p :

$$S_I(p) = \frac{1}{2} - \frac{1}{\pi} \tan^{-1} \frac{\kappa_1(p) + \kappa_2(p)}{\kappa_1(p) - \kappa_2(p)}, \quad (1)$$

where κ_1 and κ_2 are the principal curvatures of the surface, with $\kappa_1 \geq \kappa_2$. All shapes can be mapped into the interval $S_I \in [0, 1]$ and every distinct shape corresponds to a unique value of S_I , except the planar shape. Points on a planar surface have an indeterminate shape index since $\kappa_1 = \kappa_2 = 0$. For computational purposes, our implementation uses a symbolic label—an arbitrarily chosen “shape index value” of 2.0—to indicate surface planarity. The shape index of a rigid surface is not only independent of its position and orientation in space, but also independent of its scale and is unitless. Fig. 2 shows how the shape index varies in the $\kappa_1 - \kappa_2$ plane.

An object’s shape can now be characterized quantitatively in terms of its *shape spectrum*. It characterizes the shape content of an object by summarizing the area on the surface of an object at each shape index value. The shape spectrum of an object view is obtained from its range data by constructing a histogram $H(h)$ of the shape index values—we used 0.005 as the bin width—and accumulating all the object pixels that fall into each bin. The proposed shape spectrum of a view can be computed from any collection of (x, y, z) points on which the fundamental notions of metric, tangent space, curvature, and natural coordinate frames can be suitably defined. Since the shape spectrum of a view is constructed directly using the original shape index values computed at each pixel in its image, segmentation of object surfaces is avoided.

Fig. 3 shows nonplanar shape spectra (spectra computed without taking the planar points on the surfaces into account) of object views. For example, the shape spectrum of a vase, Vase2 in Fig. 3b, indicates that the main shape category present in this object is *dome*, along with a few smaller peaks in the *ridge* and *saddle-ridge* categories. Concavities in the vase are characterized by the nonzero bins below the 0.5 shape index level.

The shape spectrum derived from complete 3D surface data of an object is viewpoint-independent. However, the spectrum derived from a single range image of an object is view-dependent. The view sensitivity of this high-level feature is exploited for object view grouping as shown in Section 4. Fig. 3 also shows how shape spectra of views of various objects differ and how spectra computed from range images obtained by observing an object at nearby viewpoints are similar to one another. The strong similarities between the spectral plots of two different views of Cobra can

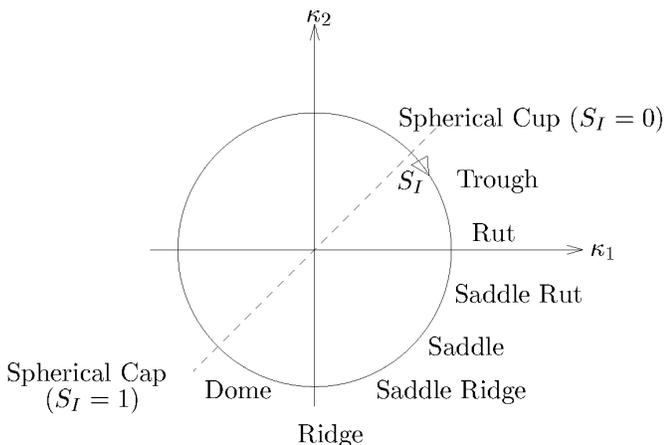


Fig. 2. The shape index (S_I) on the principal curvature ($\kappa_1 - \kappa_2$) plane.

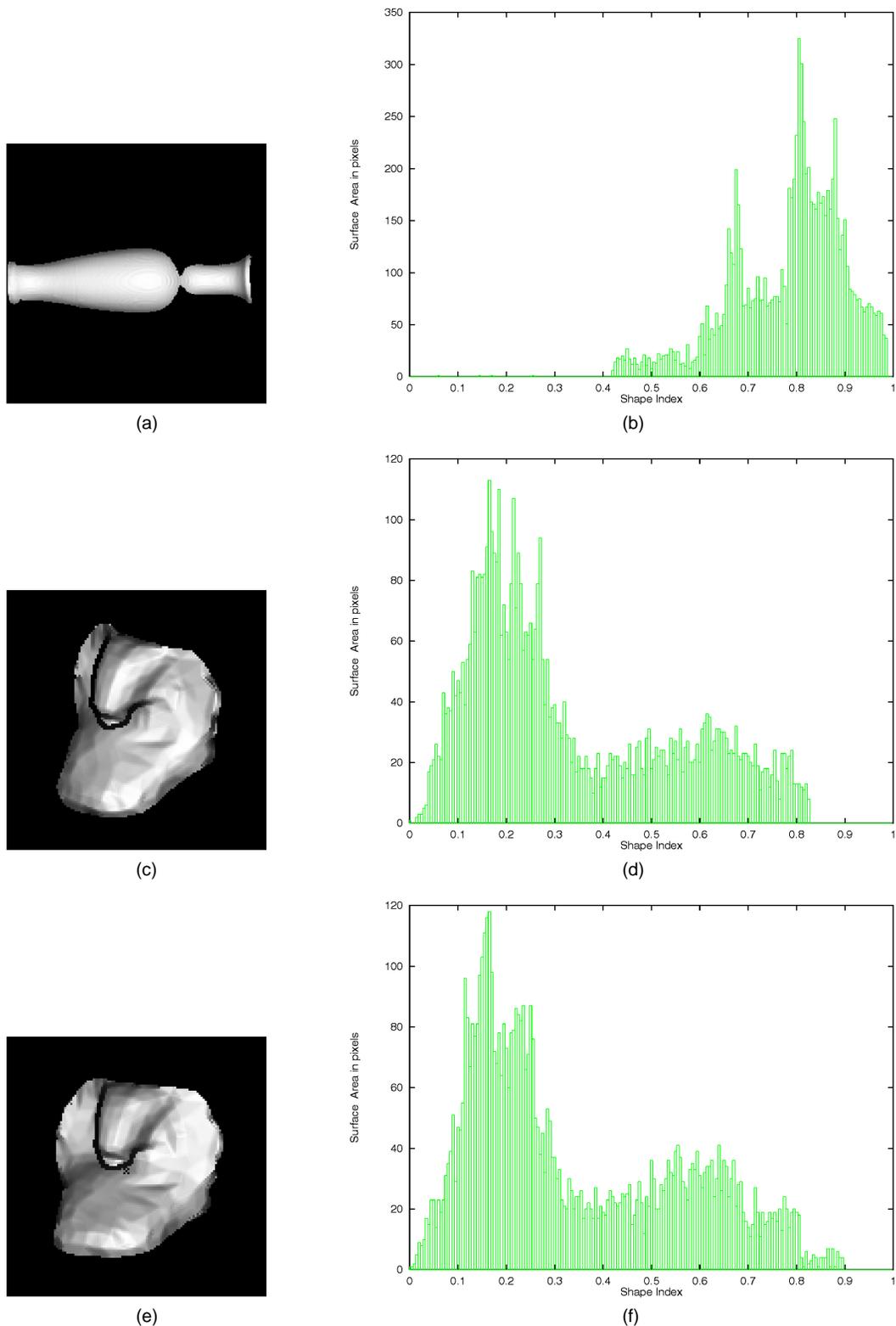


Fig. 3. Shape spectrum: (a) Range image of Vase2; (b) shape spectrum of Vase2; (c) a view of the cobra head—Cobra-1; (d) shape spectrum of Cobra-1; (e) another view—Cobra-2; (f) shape spectrum of Cobra-2.

be seen in Figs. 3d and 3f. These plots also indicate the predominance of *rut* (0.25), *ridge* (0.75), and *trough* (0.125) shapes in Cobra.

Since the spectra of purely polyhedral objects exhibit a single peak at the shape index value of 2.0 (all the planar patches contribute to this bin), it is difficult to discriminate between various views of polyhedral objects. However, shape spectrum based classification can be used to categorize object views in a database into

two classes: those that are purely planar and those that contain nonplanar shapes on the object surfaces. Since there is a huge body of techniques available for polyhedral object recognition, and since most objects in the world are free-form and nonpolyhedral, we will not discuss polyhedral object matching further. We will study mainly the use of nonplanar shape spectra for grouping object views of free-form surfaces for fast matching.

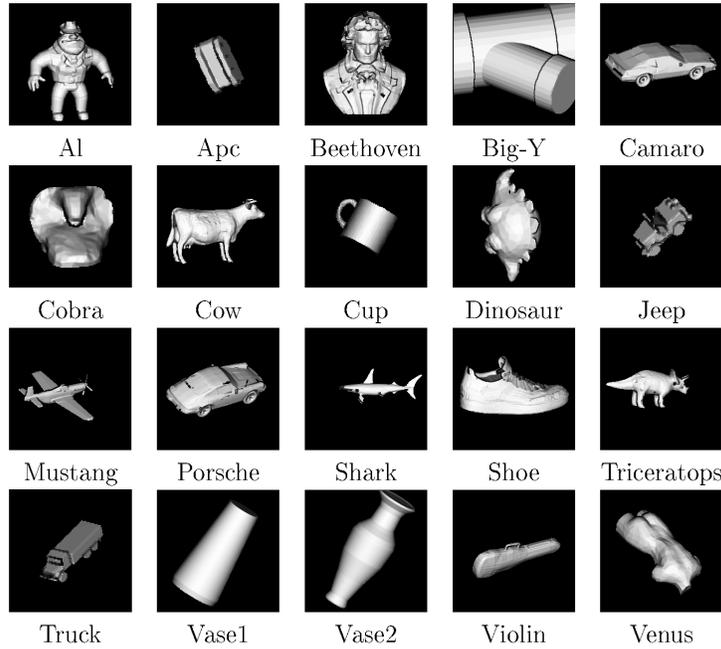


Fig. 4. Range images of 20 objects generated using 3D object models.

4 ORGANIZING OBJECT VIEWS USING SHAPE SPECTRA

We now describe how the shape spectra of object views can be used efficiently for

- 1) view grouping and
- 2) view matching.

When the model database is populated during its construction, the object identities of the views to be stored in the database are known. We first investigate whether multiple views of the same object can be clustered into meaningful groups based on their shape spectra. We have chosen to perform clustering, instead of supervised classification, to find out whether there is any inherent clustering tendency present among the training set views. Second, the object view grouping can be repeated with each set of object views, and the model database can thus be structured into a collection of distinct groups of views of each object. We propose to determine the matching efficiency and accuracy by hierarchically comparing an input view with the view cluster representatives first, followed by matching it with the views within the clusters themselves. Our primary concern is to structure a large database of object views in order to eliminate matching the input view with all the stored views, and to narrow down the possible set of views that need to be matched more comprehensively.

4.1 Feature Representation and Similarity Between Shape Spectra

A group of object views organized on the basis of “similarity” of shape spectral features would contain views that exhibit the characteristics of the same set of visible surfaces of the object. Note that views that can be obtained by rotations about the viewing direction are likely to possess similar shape spectral features and are therefore grouped together.

We have proposed a feature representation that emphasizes the spread characteristics (variance) of the spectral distribution. Our feature vector representation \mathbf{R} of a view is based on the first 10 moments of the normalized (with respect to the visible object surface area) shape spectral distribution $\bar{H}(h)$ of an object view. By normalizing the spectrum with respect to the total object area, we remove the scale (size) differences that may exist between different

objects. Moreover, we reduce the problems associated with bin quantization that cause poor performance if direct comparison of two histograms is used. These features are best understood if we observe the likeness between the shape spectrum of an object view and a probability density function of a random variable.

The 10 moments are defined as follows:

$$m_1 = \sum_h (h)\bar{H}(h); \quad m_p = \sum_h (h-m_1)^p \bar{H}(h), \quad 2 \leq p \leq 10. \quad (2)$$

Then the feature vector is denoted as $\mathbf{R} = (m_1, m_2, \dots, m_{10})$. Note that the range of each of these moments is $[-1, 1]$.

Let $O = \{O^1, O^2, \dots, O^n\}$ be a collection of n 3D objects whose views are present in the model database, \mathcal{M}_D . The j th view of the i th object, O_j^i in the database is represented by $\langle L_j^i, \mathbf{R}_j^i \rangle$, where L_j^i is the object label, and \mathbf{R}_j^i is the shape spectral moment vector.

Given a set of object representations $\mathcal{R}^i = \{\langle L_1^i, \mathbf{R}_1^i \rangle, \dots, \langle L_m^i, \mathbf{R}_m^i \rangle\}$ that describe m views of the i th object, the goal is to derive a partition of the views, $\mathcal{P}^i = \{C_1^i, C_2^i, \dots, C_k^i\}$ (see Fig. 5). Each cluster in \mathcal{P}^i contains views that have been adjudged similar, based on the dissimilarity between the corresponding moment features of the shape spectra of the views. The measure of dissimilarity between \mathbf{R}_j^i and \mathbf{R}_k^i is defined as

$$\mathcal{D}(\mathbf{R}_j^i, \mathbf{R}_k^i) = \sqrt{\sum_{l=1}^{10} (R_{jl}^i - R_{kl}^i)^2}, \quad (3)$$

where R_{jl}^i is the l th moment of the j th view of the i th object. Since the moment terms are unitless, the different moment terms can be added up coherently.

4.2 View Grouping

In order to provide a meaningful categorization of views of the object O^i , views are clustered based on their dissimilarities $\mathcal{D}(\mathbf{R}_j^i, \mathbf{R}_k^i)$ using a hierarchical clustering scheme, such as the complete-link algorithm [17]. The partition \mathcal{P}^i is obtained by split-

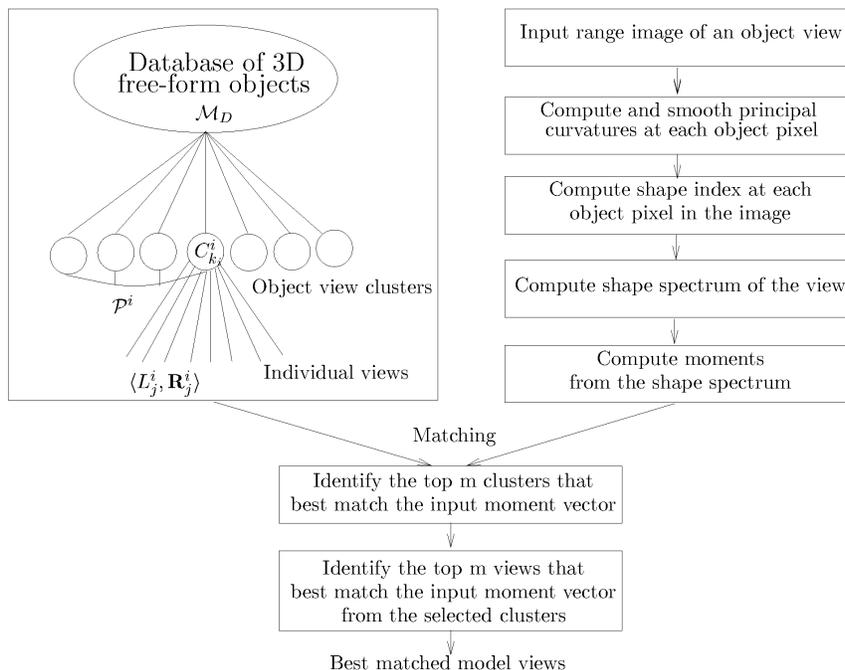


Fig. 5. Model view selection with the view grouping and matching system.

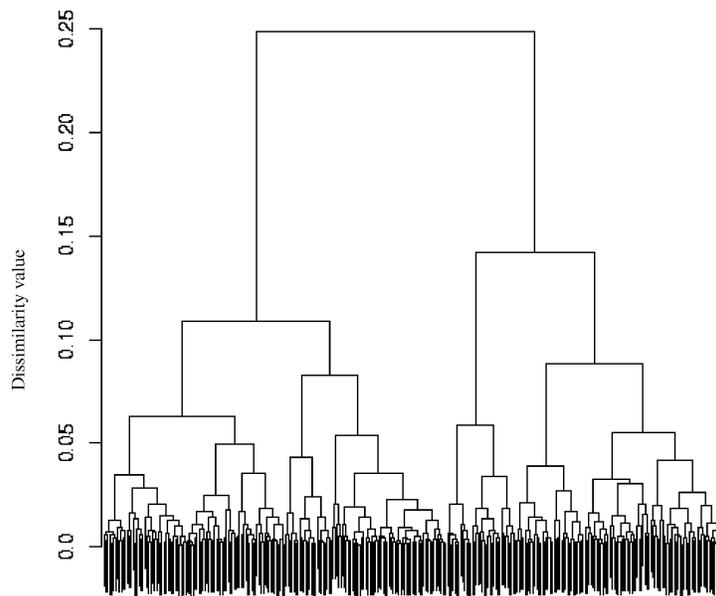


Fig. 6. Hierarchical grouping of 320 views of Cobra.

ting the hierarchical grouping of O^i at a specific level of dissimilarity in the dendrogram. The split level is chosen at a user-specified dissimilarity value to result in a set of compact and well-separated clusters. Alternatively, if the number of resultant clusters is prespecified as a design criterion, then the cut level can be automatically selected.

Once the partition P^j is determined from the training views of O^i , the database M_D is organized into a two-level structure, $M_D = \{P^1, \dots, P^n\}$, where each P^j is itself a set of view clusters. A summary representation, such as the centroid, is abstracted for each view cluster C_j^i from the moment vectors of its constituent views. Given an input view, its object label and best-matching view are

identified quickly and accurately in two stages:

- 1) The object identity is established by first comparing the moment vector of the input view with the cluster summary representations and selecting the best-matched cluster;
- 2) Comparison of the input view with the moment vectors of the views in the best-matched cluster determines the view that matches most closely with the input.

5 EXPERIMENTAL RESULTS

We have used two databases in our experiments:

- 1) a database containing 6,400 synthetically generated range images of 20 sculpted objects with 320 views per object, and

TABLE 1
OBJECT MATCHING ACCURACY WITH AN INDEPENDENT TEST SET OF 2,000 VIEWS

Object class	Correct object view classification (%) when K best-matched clusters were examined							
	$K = 1$	$K = 2$	$K = 5$	$K = 10$	$K = 15$	$K = 20$	$K = 25$	$K = 30$
Vase2	30.0	43.0	79.0	98.0	100.0			
Vase1	27.0	51.0	84.0	97.0	97.0	97.0	97.0	97.0
Big-Y	3.0	7.0	35.0	68.0	92.0	100.0		
Cobra	61.0	84.0	100.0					
Cup	31.0	56.0	90.0	98.0	100.0			
Apc	21.0	44.0	97.0	100.0				
Jeep	25.0	48.0	94.0	100.0				
Truck	17.0	48.0	88.0	98.0	98.0	99.0	99.0	99.0
Al	25.0	51.0	86.0	99.0	99.0	99.0	99.0	99.0
Beethoven	33.0	63.0	91.0	99.0	100.0			
Cow	29.0	62.0	96.0	100.0				
Dinosaur	23.0	46.0	70.0	92.0	98.0	99.0	99.0	99.0
Porsche	16.0	44.0	75.0	98.0	99.0	100.0		
Shark	36.0	46.0	80.0	93.0	96.0	96.0	98.0	99.0
Shoe	39.0	59.0	78.0	91.0	95.0	100.0		
Triceratops	23.0	41.0	77.0	97.0	99.0	100.0		
Venus	26.0	46.0	81.0	100.0				
Violin	66.0	95.0	99.0	100.0				
Camaro	21.0	26.0	66.0	97.0	99.0	99.0	100.0	
Mustang	19.0	23.0	47.0	77.0	85.0	91.0	96.0	100.0

- 2) a second database of 100 range images of 10 free-form objects (10 views per object) collected using the laser range scanner.

Fig. 4 shows the 20 complex objects, each of which was modeled using 320 different views. The range images of the objects from 320 possible viewpoints (determined by the tessellation of the viewsphere using the icosahedron) were synthesized either from the CAD models when available or from the hand-constructed object triangulations. The polyhedral models of some of the objects were collected from a public domain database (<http://www.eecs.wsu.edu/~flynn>) on the Internet.

5.1 View Clustering Results

For each object view, the shape index was computed at each pixel in its range image from the principal curvatures that were reliably estimated using an iterative curvature smoothing algorithm. Estimation of local curvatures at each surface point and the iterative smoothing of curvatures took between five and 15 minutes on the average on images of size 240×240 containing about 15,000 surface points on a SPARCstation 20. Computation of shape index at each pixel in the image, construction of the shape spectrum from S_i values, and computation of the moment feature vector took about a few seconds on the average. We clustered the views of each object based on the dissimilarity measure \mathcal{D} between their moment vectors using the complete-link hierarchical clustering scheme [17]. These steps are summarized in Fig. 5.

The dendrogram depicting the hierarchical grouping obtained with 320 views of the Cobra object is shown in Fig. 6, where the leaf nodes are the views themselves. The view grouping hierarchies of the other objects are similar to the dendrogram in Fig. 6. These clusterings demonstrate that the views of each object fall into several distinguishable clusters. The hierarchical grouping obtained from each object was then cut at a dissimilarity level of 0.1 or less to result in compact view clusters. The centroid of each of these clusters was determined by computing the mean of the moment vectors of the views falling into the cluster. Observe that the shape spectra of the views do not change with a rotation of the views about a single axis, and this leads to a more concise method for grouping multiple views.

5.2 View Matching Results

The goal of our experiments is to examine how view grouping facilitates matching in terms of classification accuracy and the number of matches necessary for correct classification of views. For these experiments, the database containing views of different objects was organized into a two-tiered structure: the first level containing all the view groups obtained from clustering views of each object individually, and the second level consisting of the views themselves in these clusters. Given a set of test range images, we studied the number of clusters that had to be examined in order to attain several levels of view misclassification rates.

We conducted several experiments in the *substitution* mode, where we used the training views themselves as test patterns and verified that object views were grouped into compact and homogeneous view clusters, thus demonstrating the discriminatory power of the shape spectrum based feature representation. These results also indicated that the simple centroid based generalization that we adopted is a reasonable scheme, and the clusters are compact enough that, after a test view picks a view cluster, it very rarely matches with a wrong view within the cluster. Each test moment pattern took about 20 ms to be correctly classified on a SPARCstation 20. We refer the reader to [18] for details.

5.2.1 Testing with 6,400 Object Views

During the *testing* phase, we trained the view grouping system with 6,400 training views (320 views per object) and tested it with 2,000 independent test views (100 per object). At the top level, the database contained 229 view clusters of objects. The second level contained the training views themselves. Each of the 2,000 test views was used as a query view, and the number of best-matched clusters that had to be examined in order to correctly identify the object class of the query view was recorded.

Table 1 summarizes the results. It can be observed that only 10 (about 4.4 percent) of the view clusters had to be examined to obtain an accurate classification of 95 percent of the test views. Complex free-form objects, such as the Cobra, Vase2, Beethoven, Cow, Violin, Venus, etc., required fewer clusters (15 top clusters or less) to be examined to obtain 100 percent correct classification of the test views, drawn from their object categories. Model views of vehicles, e.g., from the Porsche and Camaro categories, were often retrieved as the best-matched views for test views from either of

these two categories. It can be observed from the table that the number of best-matched clusters that must be examined for 100 percent object classification accuracy depends on the size of the database. These results further demonstrate that the shape spectrum based moment vector for view representation can serve as a useful pruning primitive during matching with a model database containing many complex free-form objects. Only 20 percent of the database was matched for view classification, on the average, over 2,000 test views, even when the top 30 clusters were examined.

5.2.2 Model View Selection with Real Range Data

The shape spectrum based matching scheme was also tested on real range images of free-form objects obtained using the Technical Arts White scanner in our laboratory. The views in a database of 100 range images of ten free-form objects (10 views per object) were randomly separated into two different categories:

- 1) a model database containing 50 views with five views obtained from each of the 10 different objects, and
- 2) an independent test set containing 50 views.

Fig. 1 shows the range images of the test views of the objects.

The model database was structured into two levels: At the top level were 10 view clusters, with each cluster containing five views from its object class at the second level. Each test view was first matched with the 10 view clusters to rank the best-matched three view clusters, and then the views falling into these three view clusters were examined clusterwise to select a best-matched view from each of the three clusters. This resulted in a view classification accuracy of 92 percent, with only four out of 50 test views failing to select even a single model view from their correct object classes among their top three matches. When five best-matched clusters were examined to select the best view from each one of them, the accuracy increased to 98 percent, with only one test view incorrectly classified. The wrongly classified test view was a view belonging to the Creamer class. The shapes of the surfaces visible in the wrongly classified view of Creamer were shared by views from other objects, leading to an incorrect classification. The average number of view comparisons performed in retrieving the top five hypotheses that matched a test view was 35, which is smaller than the number of view comparisons required when a linear matching of the test view with all the 50 model views is performed. Due to the relatively small size of the database (five views per object), the percentage comparisons for accurate classification of a test view is higher than with the experiments reported in Section 5.2.1. More sample views in each object class in the model database are needed to increase the discrimination between the object shapes visible in the views. Two of our examples (the Cobra head and the Phone) demonstrate a small amount of self-occlusion, and the shape spectrum successfully tolerated this.

6 DISCUSSION

We informally studied the efficacy of the moment features derived from the shape spectra of object views in classifying an input view correctly and found that only the first four moments significantly contributed to correct classification of the input. The higher order moments were low in magnitude and did not add much to the Euclidean distances computed for comparison of moment vectors. Alternative metrics, especially the Mahalanobis distance, could be used instead to compare the feature vectors and measure the similarity of views. The Mahalanobis distance ensures weighted contributions of individual feature values to the distance computed between views. A thorough comparison can be made between these two distance measures to determine the utility of the high-order moments in the feature vectors derived from the shape spectra of views.

A future research direction is to add more levels to the hierarchical database structure. For example, a set of object views can be organized based on their shape spectra into several categories: those that exhibit planar patches alone, and those that exhibit other shapes in addition to planar patches. Given this broad organization, a fine-grain organization of the latter category into views that contain purely nonconvex shapes and views that contain purely convex shapes can also be obtained. Given an input object view, its shape spectrum can be computed easily, and, then, by descending through this hierarchy, it can be compared with only a small subset of views that are likely to match best with it.

7 SUMMARY

We have addressed the problem of constructing view clusters of free-form objects. By exploiting view grouping in model databases, a small number of plausible correct matches can be quickly retrieved for more refined matching. We have proposed a novel shape spectral feature based scheme for grouping views that obviates surface segmentation and edge detection. These features allow object views to be grouped meaningfully in terms of the shape categories of the visible surfaces and their surface areas. The proposed approach is general and relatively easy to use. We have demonstrated that, in a database containing 229 view clusters of 20 sculpted objects, only the top 14 percent of the best-matched clusters need to be examined for 100 percent recognition accuracy. With the database containing 6,400 views of 20 objects, only 20 percent of the database was examined, on the average, over 2,000 independent test views for correct classification. We also demonstrated the effectiveness of the shape spectral matching scheme on real range images of views of free-form objects.

ACKNOWLEDGMENTS

We thank the reviewers for their thoughtful suggestions for improvement.

REFERENCES

- [1] P.J. Besl, "The Free-Form Surface Matching Problem," *Machine Vision for Three-Dimensional Scenes*, H. Freeman, ed., pp. 25-71. Academic Press, 1990.
- [2] C. Dorai and A.K. Jain, "COSMOS—A Representation Scheme for Free-Form Surfaces," *Proc. Fifth Int'l Conf. Computer Vision*, pp. 1,024-1,029, Boston, June 1995.
- [3] S. Chen and H. Freeman, "Computing Characteristic Views of Quadric-Surfaced Solids," *Proc. 10th ICPR*, pp. 77-82, Atlantic City, N.J., 1990.
- [4] J.J. Koenderink and A.J. van Doorn, "The Internal Representation of Solid Shape with Respect to Vision," *Biological Cybernetics*, vol. 32, pp. 211-216, 1979.
- [5] J.H. Stewman and K.W. Bowyer, "Aspect Graphs for Convex Planar-Face Objects," *Proc. IEEE Workshop Computer Vision*, pp. 123-130, Miami Beach, 1987.
- [6] Z. Gigus, J. Canny, and R. Seidel, "Efficiently Computing and Representing Aspect Graphs of Polyhedral Objects," *Proc. Second IEEE Int'l Conf. Computer Vision*, pp. 30-39, Tarpon Springs, 1988.
- [7] D.J. Kriegman and J. Ponce, "Computing Exact Aspect Graphs of Curved Objects: Solids of Revolution," *Proc. IEEE Workshop Interpretation of 3D Scenes*, pp. 116-122, Austin, 1989.
- [8] W.H. Plantinga and C.R. Dyer, "Visibility, Occlusion, and the Aspect Graph," *Int'l J. Computer Vision*, vol. 5, pp. 137-160, 1990.
- [9] D. Eggert and K. Bowyer, "Computing the Perspective Projection Aspect Graph of Solids of Revolution," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 2, pp. 109-128, Feb. 1993.
- [10] K. Ikeuchi, "Generating an Interpretation Tree from a CAD Model for 3D-Object Recognition in Bin-Picking Tasks," *Int'l J. Computer Vision*, vol. 1, pp. 145-165, 1987.
- [11] K. Sengupta and K.L. Boyer, "Organizing Large Structural Modelbases," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 4, pp. 321-332, Apr. 1995.

- [12] C. Hansen and T. Henderson, "CAGD-Based Computer Vision," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 11, pp. 1,181-1,193, Nov. 1989.
- [13] D. Eggert et al., "The Scale Space Aspect Graph," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1,114-1,130, Nov. 1993.
- [14] S.J. Dickinson, A.P. Pentland, and A. Rosenfeld, "3-D Shape Recovery Using Distributed Aspect Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 174-198, Feb. 1992.
- [15] J.B. Burns and L.J. Kitchen, "Rapid Object Recognition from a Large Model Base Using Prediction Hierarchies," *Proc. 1988 DARPA Image Understanding Workshop*, pp. 711-719, 1988.
- [16] M. Swain, "Object Recognition from a Large Database Using a Decision Tree," *Proc. DARPA Image Understanding Workshop*, pp. 690-696, 1988.
- [17] A.K. Jain and R.C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, N.J.: Prentice Hall, 1988.
- [18] C. Dorai, "COSMOS: A Framework for Representation and Recognition of 3D Free-Form Objects," PhD thesis, Dept. of Computer Science, Michigan State Univ., May 1996.