A New Paradigm for Recognizing 3-D Object Shapes from Range Data

Salvador Ruiz-Correa†, Linda G. Shapiro† and Marina Meilă‡
†Department of Electrical Engineering
‡Department of Statistics
University of Washington, Seattle, WA 98105
{sruiz@u, shapiro@cs, mmp@stat}.washington.edu

Abstract

Most of the work on 3-D object recognition from range data has used an alignment-verification approach in which a specific 3-D object is matched to an exact instance of the same object in a scene. This approach has been successfully used in industrial machine vision, but it is not capable of dealing with the complexities of recognizing classes of similar objects. This paper undertakes this task by proposing and testing a component-based methodology encompassing three main ingredients: 1) a new way of learning and extracting shape-class components from surface shape information; 2) a new shape representation called a symbolic surface signature that summarizes the geometric relationships among components; and 3) an abstract representation of shape classes formed by a hierarchy of classifiers that learn object-class parts and their spatial relationships from examples.

1. Introduction

Over the past two decades, the problem of recognizing free form objects from 3-D range data scenes has been intensively studied in computer vision research due to its prominent relevance to a variety of application fields. These include, among others, robot vision, autonomous navigation, automated inspection and measurement, satellite image analysis, and more recently, retrieval of 3-D objects based on shape [15]. However, most of the successful approaches developed up to date have concentrated on designing surface shape representations for recognizing specific objects in a database, and very little attention has been paid to the more general problem of recognizing objects belonging to a particular *shape class*, a collection of 3-D objects that share a set of defining surface characteristics that are visually similar and occur in similar geometric configurations. Figure 1 shows instances of four shape classes considered in this work: human heads, rabbits, snowmen, and floppy-eared dogs.

Recognizing members of object classes from their shape

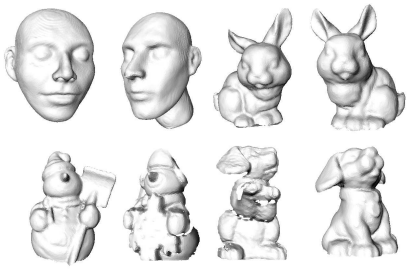

Figure 1. Instances of four shape classes: human heads, rabbits, snowmen and dogs.

is difficult for several reasons. In the first place, how to construct a quantitative description of a shape class that accounts for the complexities in the categorization process is an open question. In real applications, human perception, knowledge and judgment are used to make qualitative definitions of a class and to make distinctions among different classes. However, categorization in humans is not yet well understood, and no one knows what information is used and what kind of processing takes place when constructing categories [13]. For these reasons, learning classes of objects for recognition purposes generally occurs in a supervised setting, an ill-defined problem that must be regularized by introducing constraints [22]. In the third place, the surface shape representation is very important in a recognition procedure. Which representation is more suitable for learning shape classes is not currently known. Finally, the nature of the input data is complex. Real range scenes may contain multiple objects and the class members have to be identified amongst varying amounts of clutter. Range scenes also contain noise and occlusion so there is only incomplete information pertaining to the objects of interest.

Most of the successful 3-D object recognition systems reported in the literature use a technique called *alignment* [11] that finds sets of point correspondences between a 3-D model and a 2-D or 3-D scene using attributes of these points, such as surface signatures. A surface signature at a given point on a surface is a descriptor that encodes the geometric properties measured in a neighborhood of the point.

The correspondences are used to solve a set of linear equations that determine the parameters of a potential rigid body transformation from the model to the scene. Using this transformation, the model is projected to the scene and verification techniques are used to determine if this is a good match. The various verification techniques all compare the points of the transformed model to points of the scene to determine how well the model overlays on the scene. This type of verification is dependent on a specific object model of which there is an exact instance in the scene. It has been used very successfully in industrial machine vision, but it does not extend well to recognizing *classes* of similar objects.

In this paper we describe and empirically test a *3-D shape detector* that implements a component-based approach for recognizing members of classes of 3-D object shapes in real range scenes. Our shape detector attempts to overcome the problems inherent to alignment-based recognition systems by learning the shape-class *components* and their *spatial configuration*. The key elements of our methodology are: 1) a novel way of processing 3-D surface shape information for learning and extracting shape-class components; b) a new shape representation called a *symbolic surface signature* that encodes the spatial relationships among the class components; and c) a *shape class model* that consists of a three-level hierarchy of classifiers that learn object-class parts and their relationships from a set of *surface signatures* embedded in a Hilbert space. The first two levels of the hierarchy extract the components. The third level verifies their geometric relationships. The surface shape representations combined with the hierarchy of classifiers result in an abstract representation of an object shape class that is robust to scene clutter and occlusion.

We used support vector machines (SVMs) for implementing the classifiers used by our 3-D shape detector. Specific details are described in Section 3.2. The readers interested in the basics of SVMs for pattern recognition are invited to consult [18].

The outline of the paper is as follows. We summarize the related work in Section 2. Section 3 is devoted to our symbolic surface signatures and our recognition method. Section 4 describes the experimental protocol and the results. Finally, Section 5 summarizes the key points of our investigation, open questions and future work.

2. Related Work

Free-form object recognition in range data has been a very active area of research. The existing approaches span many different axes. In early work, Faugeras and Hebert [8], and Besl [1] studied the difficulties in matching free-form objects using point, curve and surface features. Nevatia and Binford [14] used generalized cylinders to create symbolic descriptors for recognizing free-form articulated objects in the presence of occlusion. Raja and Jain [16] developed a technique for fitting and classifying deformable superquadrics to range data, where the deformations considered were tapering and bending. Surface curvature has also been used to define local descriptors of surface shape. Besl and Jain [3] used Gaussian curvature and mean curvature to classify local surface shape into basic categories such as peaks, pits, ridges and valleys. Dorai and Jain [7] defined two new curvature measures (shape index and curvedness) along with a discrete spectral extension of surface categories to build a view-dependent representation of free-form objects. Their COSMOS system uses a histogram of the shape index value to characterize the curvature of a particular view, and constructs an object database consisting of many views of each object to be recognized. Delingette *et. al.* [6] developed the spherical attribute image (SAI) for representing 3-D surfaces. The SAI representation maps points of an object surface to a tessellated sphere. Local features are stored at the vertices of the sphere that correspond to the surface points. Matching between two objects reduces to finding the best rotation aligning the scene and model SAIs.

Stein and Medioni [20] used changes in surface orientation to match local regions of surfaces. They utilized the so-called "splash" representation and a structural indexing approach to matching. Chua and Jarvis [5] developed the "point signature", a local descriptor of shape that encodes the minimum distances from points on a 3-D contour to a reference plane. The idea of "point signatures" was further developed in various investigations. These include the spin image representation of Johnson and Hebert [12], the curvature signatures of Yamani *et. al.* [23], the harmonic shape images of Zhang and Hebert [24] and the spherical spin images of Ruiz-Correa *et. al.* [17]. Recently, Osada *et. al.* [15] developed the shape signature of a complete 3-D model as a probability distribution sampled from a shape function measuring geometric properties of the 3-D model. Funkhouser *et al.* [9] extended the work on shape distributions by developing a representation of shape for object retrieval from multi-modal queries. The representation is based on a spherical harmonics expansion of the points of a polygonal surface mesh rasterized into a voxel grid aligned to the center of mass of the object. Query objects are matched to the database using a nearest neighbor classifier. Shape distributions and harmonic descriptors can operate on degenerate surface mesh models but they lack robustness to scene clutter and occlusion due to their global character.

3. Our Approach

The main contribution of our *3-D shape detector* is the novel way surface shape information and geometrical relationships are combined in a hierarchy of classifiers to form

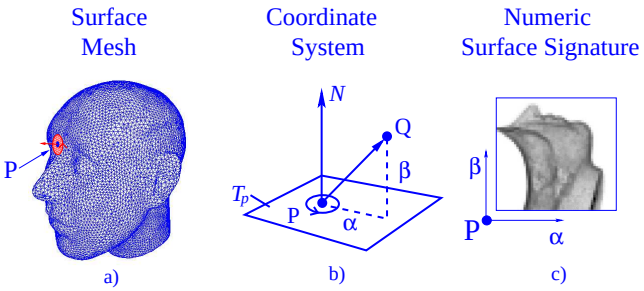

Figure 2. The numeric surface signature for point P is constructed by accumulating in a 2-D histogram the coordinates α and β of a set of contributing points (such as Q) on the mesh representing the object.

an abstract representation of an object shape class. The method utilizes three methodologies. First, a numeric surface signature representation is used for encoding local surface shape information. Secondly, our symbolic signature representation is used for encoding the geometrical relationships among the shape-class parts. Finally, the SVM technologies are used for learning the parts and their spatial dispositions.

3.1. Surface signatures

The surface signatures developed by Johnson and Hebert [12] encode the surface shape of free form objects. In contrast to the shape distributions and harmonic descriptors, they are robust against the clutter and occlusion generally present in range data. Experimental evidence has shown that the spin image and some of its variants are the preferred choice for encoding surface shape whenever the normal vectors of the surfaces of the objects can be accurately estimated [17]. We use spin images as the numeric signatures in this work.

Numeric surface signatures. A spin-image [12] is a two-dimensional histogram computed at an oriented point P of the surface mesh of an object (see Figure 2). The histogram accumulates the coordinates α and β of a set of *contributing points* Q on the mesh. Contributing points are those that are within a specified distance of P and for which the surface normal forms an angle of less than the specified size with the surface normal N of P. This angle is called the *support angle*. As shown in Figure 2, the coordinate α is the distance from P to the projection of Q onto the tangent plane T_P at point P; β is the distance from Q to this plane.

Labeled surface regions. Our symbolic surface signatures (defined below) encode the spatial relationships among a set of *shape-class components*. A shape class component is a group of connected surface mesh points whose numeric signatures are similar as defined by the clustering algorithm to be described in Section 3.2. The different components of a class can be represented on a *labeled surface mesh* (Fig. 3a); each vertex of the mesh has an associated symbolic label referencing the component in which it lies. The geometric configuration of the class components can be extracted from the labeled surface mesh using our *symbolic surface signatures*.

Symbolic surface signatures are somewhat related to numeric surface signatures in that they also start with a point P on the surface mesh and consider a set of contributing points Q, which are still defined in terms of distance from P and support angle. The main difference is that they are derived from the labeled surface mesh (shown in Figure 3a). For symbolic surface signature construction, the vector $\overline{PQ}$ in Figure 3b is projected to the tangent plane at P where a set of orthogonal axes γ and δ have been defined. The direction of the $\delta - \gamma$ axes is arbitrary, since no curvature information was used to define preferred directions. This ambiguity will be resolved by the methods described in Section 3.2. The discretized version of the γ and δ coordinates of $\overline{PQ}$ are used to index a 2D array, and the indexed position of the array is set to the component label of Q. Note that it is possible that multiple points Q that have different labels project into the same bin in the symbolic surface signature. In this case, the label that appeared most frequently is assigned to the bin. The resultant array is the symbolic surface signature at point P. The signature captures the relationships among the labeled regions on the mesh. It is shown as a labeled color image in Figure 3c.

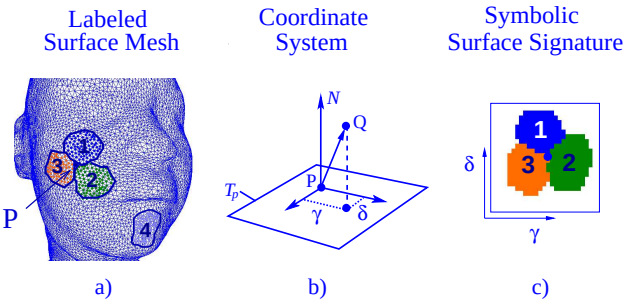

Figure 3. The symbolic surface signature for point P on a labeled surface mesh model of a human head. The signature is represented as a labeled color image for illustration purposes.

3.2. Learning shape classes

We consider the learning task for which we are given n surface meshes $\mathbf{C} = \{C_1, \cdots, C_n\}$, which are samples of the shape class $\mathcal{C}$. The problem is to use the given *training* examples to construct an algorithm that determines whether or not a new mesh C belongs to class $\mathcal{C}$. We start by assuming that the correspondences between all the mesh points of the instances in C are known. This can be achieved by using a morphable surface models technique such as the one described in [19]. We also assume the number of samples in $\mathbf{C}$ is large enough to allow SVM learning. In some applications, however, this may not be the case, and prior knowledge about the shape class must be used to create virtual

examples for training. This is not an uncommon practice when learning from examples. In some cases, the training set is enlarged in order to incorporate invariances in the class model [18]. In others, such as in learning facial components for face detection, 3-D head models are used to automatically generate faces with arbitrary pose and arbitrary illumination [10].

Before shape class learning can take place, one has to specify salient feature components associated with $\mathcal{C}$. Each component of a class is identified by a particular region located on the surface of the class members. The components are constructed according to step I of the algorithm described below. Once the class components have been found, our proposed 3-D shape detector is trained using the steps II and III of the following algorithm.

Step I . The input of this phase is a set $\mathbf{C}$ of surface meshes that are samples of object class $\mathcal{C}$. The goal is to extract a set of components associated with $\mathcal{C}$. The extraction process is performed using the following region growing algorithm.

1. Select a set of critical points on a training object for class $\mathcal{C}$; the number m of points and their locations are currently chosen manually by the experimenter.
2. Use known correspondences to find the corresponding critical points on all training instances in $\mathbf{C}$.
3. For each critical point p, compute the numeric surface signature of the corresponding points on each instance of $\mathbf{C}$. This set of signatures is the training set T_p for critical point p of class $\mathcal{C}$.
4. For each critical point p, train a *component detector* (implemented as a single class ν-SVM) to learn a component about that point, using the training set T_p. Initially, the component detector learns only the selected critical point (in each object of the training set). Then iteratively, the region about the point is grown by selecting an adjacent point, recomputing the error of the classifier, and adding the new point to the region if the new error is lower than the previous error. The error measure used [21] is given by $E = \#SV_p/k$, where $\#SV_p$ is the number of support vectors in the component detector for p, and k is the number of samples used to train the component detector.

This component-growing technique is related to the one used by Heisele [10] *et. al.* for categorizing objects in 2-D images by learning and combining object parts. An example of the components grown by this technique on a training set of 600 human heads is shown in Figure 3a. The first two objects in Figure 1 are typical examples of this training set.

At the end of step I, there are m component detectors, each of which can identify the component of a particular critical point of the object shape. That is, when applied to a 3-D mesh, each component detector will operate as a filter that determines which vertices it thinks belong to its learned component (*positive surface points*), and which vertices do not. These m classifiers are the Level-1 output of our procedure. Note that due to the nature of the data or to classification errors, more than one classifier may label a mesh point as a positive point. This is expected, since each of the m classifiers is trained individually on a particular component. The ambiguity can be resolved by training a multiple-class classifier to discriminate among components as described below.

Step II. The goal of this step is to improve the accuracy of the output labels of the component detectors, which our experiments proved to be not reliable enough. The input of this step is the training set of numeric signatures and their corresponding labels for each of the m components in $\mathcal{C}$. The labels are determined by the step-I component detectors previously applied to $\mathbf{C}$. The output is a *component classifier* (implemented as a pairwise multi-way classifier that uses binary ν-SVMs) which, when given a *positive surface point* of a 3D mesh previously processed with the Level-1 filter, will determine the particular component of the m components to which this point belongs. The component classifier is essential for the proper operation of our method.

Step III. The purpose of this step is to learn the spatial relationships of the labeled components within class $\mathcal{C}$. The input data is the set C of training meshes with each vertex labeled with the label of its component or zero if it does not belong to a component. The training proceeds as follows. 1) Select l *shape points* on a surface mesh in C that will be used to characterize the *object shape class*. (In our experiments, $l = 1$ was used, but an arbitrary number is possible.) Again use known correspondences to find the corresponding points on each training sample of C. 2) Compute symbolic surface signatures at each shape point of each training instance and form the corresponding training set. 3) Use the training set to teach a *symbolic signature detector* (implemented as a single-class ν-SVM) to determine which symbolic surface signatures are associated with the class being learned. The output is a symbolic signature detector for object class $\mathcal{C}$ that takes in a labeled mesh with associated symbolic surface signatures and decides for each point if there is evidence of the component configuration that defines class membership in class $\mathcal{C}$.

A Mercer kernel for symbolic surface signatures. In order to train a single-class SVM to recognize symbolic surface signatures, they must be embedded in a vector space. To perform the mapping in the context of SVMs, we designed a simple kernel function for measuring the similarity between symbolic signatures. The kernel is constructed as follows. Let A and B be two square matrices of dimension N storing arbitrary labels. Let $A * B$ denote a binary square matrix whose elements are defined as $[A * B]_{ij} = \mathrm{match}\left([A]_{ij}, [B]_{ij}\right)$, where $\mathrm{match}(a, b) = 1$ if $a = b$, and 0 otherwise. The symmetric mapping $< A, B >= (1/N^2)\sum_{ij}[A * B]_{ij}$, whose range is the interval $[0, 1]$, can be interpreted as the cosine of angle θ_{AB} between two unit vectors on the unit sphere lying within a single quadrant.

The angle θ_{AB} is the geodesic distance between them. Our kernel function is defined as $k(A, B) = \exp(-\theta_{AB}^2/\sigma^2)$. It allows us to embed our symbolic signatures into a Hilbert space of infinite dimension. One can design other kernel functions that take local correlations into consideration. We defer their study to a future paper.

As noted in the previous section, the symbolic surface signatures are defined up to a rotation. For this reason we use the virtual SV method for training the single-class SVM [18]. The method consist of training a single-class SVM on the symbolic surface signatures to calculate the support vectors. Once the support vectors are obtained, new virtual support vectors are extracted from the labeled surface mesh in order to include the desired invariance transformation; that is, a number r of rotated versions of each support vector is generated by rotating the coordinate system used to construct each symbolic signature (see Fig. 3). Finally, the single-class SVM used in the recognition algorithm is trained with an enlarged data set consisting of the original training data and the set of virtual support vectors.

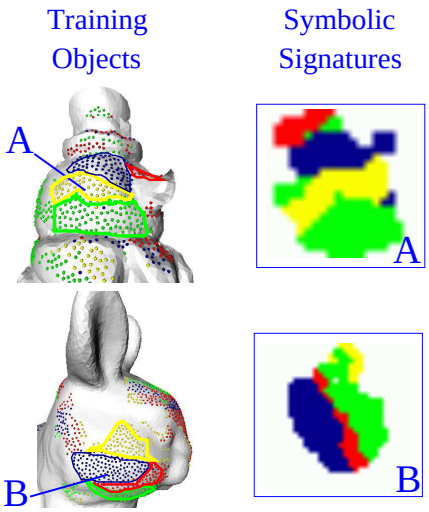

Figure 4. Components grown by our technique on sample training instances of the snowman and rabbit classes. The symbolic signatures for points A and B form a 2-D representation of these components.

3.3 Recognizing shape class members

Figure 4 illustrates two examples of the components produced by our method on a snowman and a rabbit. The figure shows the components on the actual training objects used in the experiments to be described in Section 4. Figure 5 illustrates the recognition procedure for the snowman class. Given an input surface mesh from a test scene, the component detector and component classifier of the learned model are applied to the numeric surface signatures of its points (Fig. 5a). Boundary points are excluded since their normal vectors cannot be accurately estimated. The output of this phase is the labeled surface mesh shown in Fig. 5b. The labeled mesh is further processed by applying a connected components algorithm. Each connected component consist of groups of labeled points connected by a path on the mesh. Components with less than 5 points are filtered out. The symbolic signature detector is then applied to the symbolic surface signatures of each of the points of the filtered mesh. The symbolic signatures of four such points are shown in Fig. 5c. The output of this step is a new labeled surface mesh. The label assigned to a point of the new mesh is set to 1 if it belongs to the learned symbolic signature class, and 0 otherwise. The connected components algorithm is applied and the resulting components are filtered again based on their size. Components with less than 3 elements are discarded. The remaining components correspond to regions in the scene where the shape points of the models are likely located (Fig. 5d).

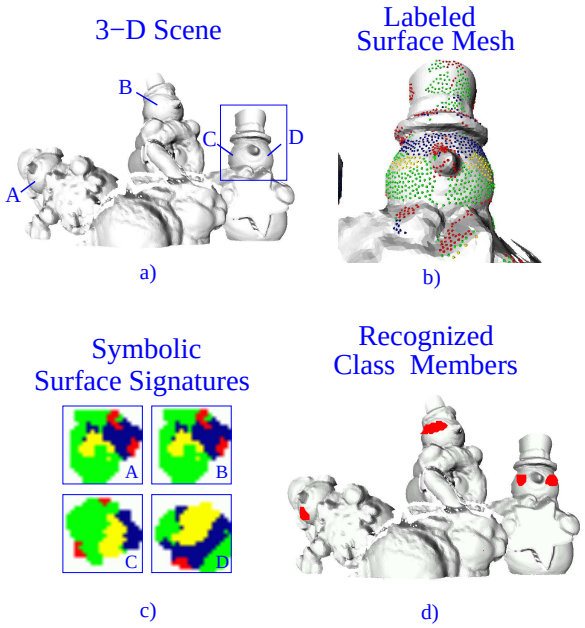

Figure 5. Recognition Example. a) 3-D scene containing three members of the snowman class. b) Labeled surface mesh (detail) obtained after processing the numeric signatures of the original scene with the first two levels of classification. The labels of the mesh are shown as small colored spheres centered in the vertexes of the mesh. c) Symbolic surface signatures of points A, B, C, and, D. The signatures are represented as color images for illustration purposes. d) Recognition results. The red blobs indicate the regions in the scene where the shape class model associated with the snowman was found.

4 Experiments

We developed an experimental protocol to validate our algorithm. The protocol consisted of three recognition tasks

Figure 6. Recognitions examples. The first two snowmen on the first row are training objects. The others are test objects recognized by our algorithm. All except the last three are real objects.

of increasing level of difficulty that allowed us to investigate the generalization ability of our method on the one hand, and its performance with respect to clutter and occlusion, on the other. The first task considered scenes with a single class member and low levels of occlusion ($< 55\%$). The second task considered scenes with multiple objects containing relatively low clutter and occlusion. The third task included scenes with large values of occlusion and clutter.

Data sets and training. Three object classes were considered for learning: snowmen, rabbits, and dogs. Two real objects for each class were used for initializing the training sets. Eighteen objects were used to create the testing sets (14 snowmen, 2 rabbits and 2 dogs). Training and testing data sets were enlarged by applying local and global deformations to the original models. The enlarged data set contained 600 models for each class. The global deformations included 8 parametric morphings such as tapering, twisting, and bending, and each deformed model included at least 5 different morphings. The morphings and their respective parameters were randomly selected from a specific range. The local deformations were produced by constructing elastic models of the original two objects and applying small random forces to their surfaces using a multi-resolution approach [4]. For all the experiments described below, the numeric signatures had the following parameters: bin size 1 (mm), image width 40, and support angle 60^o. For the symbolic signatures the bin size was 1 (mm), the image width 20, and support angle 30^o. The resolution of all the surface meshes in the study was set to 1 (mm). All the SVMs used by our method were trained using a parameter $\nu = 0.05$ [18]. We also used a Gaussian kernel function for the component detectors and component classifier. The parameter σ of the kernel functions was selected using cross validation. The number of rotations r for training the novelty detector that learns the symbolic signatures was set to 10. For each shape class model, a total of $72,000$ numeric signatures were considered for training the first two levels of the classifier hierarchy. In turn these classifiers generated $3,600$ labeled surface models from which $11,400$ symbolic surface signatures were extracted for training the third classifier.

The construction of shape class models is a time consuming task. It takes about two days to produce a single class model. The process includes the construction of the surface meshes, the generation of training and testing sets, and the training of the classifiers. Each hand-made clay snowman took about 3 hours to create, and the realistic human heads took two days each. Collecting and preprocessing each of the scenes in the database took about 45 minutes per scene.

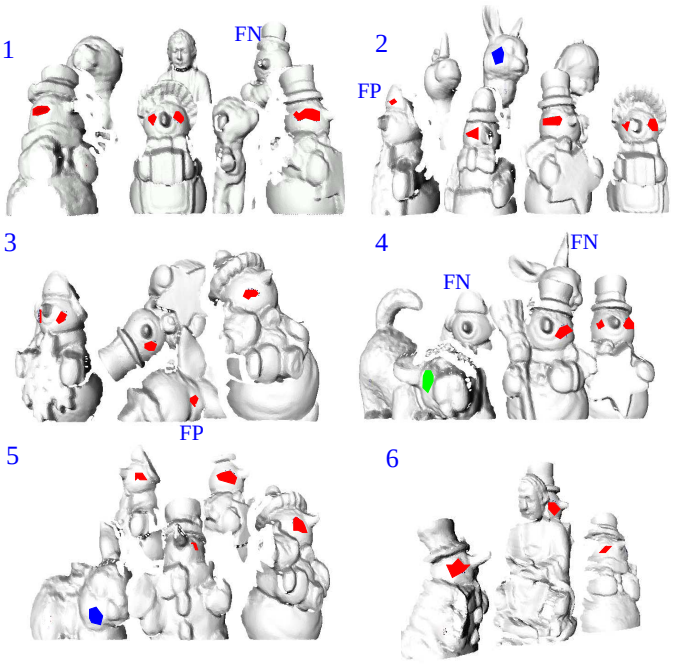

Figure 7. Recognitions examples for several typical 3-D scenes in our database. The regions where the snowmen were recognized are highlighted in red, rabbits in blue, and dogs in green.

Task 1 (generalization performance). A total of 580 experiments using scenes containing a single object were conducted to test the generalization performance of the algorithm. Four hundred scenes had complete virtually de-

formed objects. One hundred and sixty contained real objects in arbitrary pose. The algorithm was able to categorize previously unseen real test objects including 14 snowmen, 2 dogs and 2 rabbits. The recognition rate obtained for these experiments was 95.2%. Figure 6 shows ten of the actual snowmen recognized in this task.

Tasks 2 and 3 (recognition in complex scenes). A total of 50 3-D objects were used in the study. Our test database consisted of 510 range images created by placing, without any systematic method, two to six models in the 3-D scene by hand. Each scene always contained at least one member of the learned classes. For each of the 510 scenes, the recognition state was classified as a *true positive* if the algorithm detected a class member in the scene; a *false positive* if the algorithm labeled an object a member of a class to which did not belong, or if the region found did not match the location of the shape points; a *true negative* if a non-class member was judged as such; and a *false negative* if a member of a class was not recognized. The level of clutter and occlusion were measured using the methods described in [17]. In task 2 the recognition algorithm was executed on all the scenes with levels of clutter and occlusion smaller than 70% and 65%, respectively (about 75% of the total). In task 3, all the scenes in the database were processed.

Qualitative Results. Figure 7 shows the recognition of class members in typical 3-D scenes in our database. The colored blobs indicate the region of the scene where the class members were found. From the views it is clear that the models are closely packed, a condition that creates a cluttered scene with occlusions. However the proposed algorithm is able to recognize most of the class members. Some objects were not recognized because insufficient data was present. Scenes 1 and 4 contain false negatives for the snowman and rabbit classes. The level of occlusion for the unrecognized models was greater than 75% in all cases. For the same models, the level of clutter was 68.26% 69.03%, and 70.24%, respectively. Scenes 2 and 3 have one false positive result each. A plausible reason for the presence of false positives is the fact that a single class model is used for recognition. This model was constructed around relatively close critical points on the surface of the training objects in order to minimize the effects of the clutter and occlusion. For example, in scene 3 the rounded back of an angel's head was interpreted as a snowman head.

Quantitative Results. Table 1 summarize the average performance of our algorithm. For task 2, the recognition rate is roughly 90%. These are reasonable measures considering that in reality, the objects in a scene are not as packed as they are in scenes in our database. The recognition rate for task 3 was roughly 85%. This rate is relatively high considering the high levels of occlusion and clutter present. The false positives rate is relatively high in both cases. Finally, the average CPU time for the classifiers to process a single correspondence was 21.32 ± 0.002 (ms) for levels 1 and 2, and 75.78 ± 0.013 (ms) for level 3, on a 2.0 GHz Pentium 4 processor. The average processing time for the scenes in our database was roughly 12 min mainly due to unoptimized code. Optimization should reduce the average processing time in half.

	TASK 2				TASK 3			
C	RR	FN	FP	TN	RR	FN	FP	TN
S	91	9	31	69	87.5	12.5	28	62
R	90.2	9.8	27.6	72.4	84.3	15.7	24	76
D	89.6	10.4	34.6	65.4	88.12	11.88	22.1	77.9

Table 1. Algorithm performance (%) for recognition tasks 2 and 3. Key: (C) class, (RR) recognition rate, (FP) false positives rate, (TN) true negatives rate, and (FN) false negatives rate. (S) snowman, (R) rabbit, and (D) dog.

5. Conclusions and Future Work

We have presented a new paradigm for recognizing members of classes of 3-D shapes. Our approach contains three new processes: 1) extracting shape-class components from mesh representations of free form 3-D objects using a novel component-growing approach based on a bound on the expected probability error of a single-class classifier; 2) representing the spatial arrangement of class components by 2-D symbolic signatures; 3) abstracting a shape class from a hierarchy of classifiers. Preliminary experiments have been performed on three shape classes, snowman, rabbit and dog, with promising results. The method had a high recognition rate and was able to generalize, even in the presence of significant scene clutter and object occlusion.

Despite these encouraging results, there are issues to investigate. 1) Our method is able to model shape classes containing significant shape variance and can absorb about 20% of scale changes. A multiresolution approach could be used for applications that require full scale invariance. 2) The algorithm described uses a single shape class model to recognize class members. The shape class model is constructed around critical points that are relatively close to each other in order to minimize the effect of clutter and occlusion. This makes the algorithm prone to false positives. This could be alleviated by utilizing several shape models on the same object class to make a decision. Multiple shape points could also be used. 3) The selection of critical and shape points for making the models is also an issue. We are investigating ways of making the selection of salient features semi-automatic. 4) Finally, the method has been tested only on clay models. We would like to apply it to a real life problem and plan to attempt to recognize classes of genetically-caused deformations in human faces. These craneofacial pathologies can be reproduced using the same

techniques employed for creating the virtual models in our training sets.

References

[1] P. J. Besl, *The free form surface matching problem* in Machine Vision for Three-Dimensional Scenes, pp 25-71, Academic Press, San Diego, 1990

[2] P. J. Besl and R. Jain, "Range Image Understanding," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 430-449, 1985.

[3] P. J. Besl and N. D. McKay, "A method of registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14, 1992, pp. 430-449.

[4] S. Capell, S. Green, B. Curless, T. Duchamp, and Z. Popovic. "A Multi-resolution Framework for Dynamic Deformations," Proceedings of the 2002 ACM SIGRAPH.

[5] C. S. Chua and R. Jarvis, "Point Signatures: A New Representation for 3-D Object Recognition," *International Journal of Computer Vision*, Vol. 25, No. 1, 1997, pp. 63-85.

[6] H. Delingette, M. Hebert, and K. Ikeuchi, "A spherical representation for the recognition of curved objects ," *Proc. IEEE Int. Conf. On Computer Vision*, Berlin Germany, May 1993, pp. 103-112.

[7] C. Dorai and A. K. Jain, "COSMOS-A Representation Scheme for 3-D Free-Form Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10), October 1997, pp. 1115-1130.

[8] O. D. Faugeras, M. Hebert, "The Representation, Recognition, and Localization of 3-D Objects," *The International Journal of Robotics Research*, Vol. 5, No. 3, 1986, pp. 27-52.

[9] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs "A Search Engine for 3D Models," *ACM Transactions on Graphics*, 22(1), pp. 83-105, January 2003.

[10] B. Heisele, T. Serre, M. Pontil, T. Vetter and T. Poggio. "Categorization by Learning and Combining Object Parts," *In: Advances in Neural Information Processing Systems*, 14, Vancouver, Canada, Vol. 2, 1239-1245, 2002.

[11] D. P. Huttenlocher and S. Ullman, "Recognizing solid objects by alignment with an image," *International Journal of Computer Vision*, Vol 5, No.2, 1990, pp. 1995-212.

[12] A. E. Johnson and M. Hebert, "Using Spin Images for Efficient Object Recognition in Cluttered 3D scenes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(5), pp. 433-449, 1999.

[13] D. L. Medin, C. M. Aguilar, *Categorization*. In R.A. Wilson and F. C. Keil (Eds.). The MIT Encyclopedia of the Cognitive Sciences, Cambridge, MA, 1999.

[14] R. Nevatia and T. O. Binford, "Description and recognition of curved objects," *Artif. Intell.*, 8, 1977, pp. 69-76.

[15] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Matching 3-D models with shape distributions," *Shape Modeling International*, 2001, pp. 154-166.

[16] N. S. Raja and A. K. Jain, "Recognizing geons from superquadrics fitted to range data," *Image Vision Comput.*, 10, 1992, pp. 179-190.

[17] S. Ruiz-Correa, L. G. Shapiro, and M. Meilă, "A New Signature-based Method for Efficient 3-D Object Recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2001*, Vol. 1, pp. 769 -776.

[18] B. Scholköpf and A. J. Smola, *Learning with kernels*. MIT Press, Cambridge Massachusetts, 2002.

[19] C. R. Shelton "Morphable Surface Models," *International Journal of Computer Vision*, 38(1), 75-91, 2000.

[20] F. Stein and G. Medioni, "Structural Indexing: Efficient 3-D Object Recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(2), 1992, pp. 125-145.

[21] D. M. J. Tax and R. P. W. Duin, "Uniform Object Generation for Optimizing One-class Classifiers," *Journal of Machine Learning Research*, 2(2001), pp. 153-171.

[22] V. N. Vapnik, *Statistical Learning Theory*. John Wiley and Sons., New York, 1998.

[23] S. M. Yamani, A. A. Fraag, and A. El-Bialy, "Free-form surface registration and object recognition using surface signatures," *IEEE Int'l Conf. Computer Vision 1999*, Kerkyra, Grece, 1999.

[24] D. Zhang and M. Hebert, "Harmonic maps and their applications in surface matching," *IEEE Conference in Computer Vision and Pattern Recognition 1999*, Vol 2, pp. 530.