A Symbolic Representation for 3-D Object Feature Detection

Pamela J. Neal
Department of Electrical Engineering
U.S. Air Force Academy, Colorado
pamela.neal@usafa.af.mil

Linda G. Shapiro
Department of Computer Science
University of Washington
Seattle, Washington
shapiro@cs.washington.edu

Abstract

In this paper we define a spatial symbolic model that can be used to describe classes of 3-D objects (anatomical and man-made) and a method for finding correspondences between the features of the symbolic models and point sets of 3-D mesh data. An abstract symbolic model is used to describe spatial object classes in terms of parts, boundaries, and spatial associations. A **working model** *is a mechanism to link the symbolic model to geometric information found in a sensed instance of the class, represented by a 3D mesh data set. Matching is performed in a three-step procedure that first finds* **working sets** *of points in the mesh, then fits* **constructed features** *to these sets, and finally selects a subset of these constructed features that best correspond to the features of the working model.*

1. Introduction

The goal of this paper is to develop a new method to match a symbolic 3D object model that represents a generic class of objects to sensed 3D data from an instance of that class. Even if the identity of the object is already known, identifying the occurrence of the features that are known to be present in such an object can be useful for locating the object in 3D space and for providing named landmarks to be used in further processing. Location and identification of such features can be used in a variety of applications, such as object reconstruction, telerobotic operations, and medical informatics.

Use of symbolic models of 3-D objects is often found in the field of artificial intelligence. The models are generally used in reasoning, and sometimes lack a real connection to a physically existing object [4]. Our model has two levels; a purely symbolic level that is useful for reasoning about an object and an intermediate level which contains physical constraints that enable us to tie real object data to the high-level model. In this way, we are able to use reasoning about the object to identify the real features in the mesh representation of the object.

In computer vision, *parameterized models* are most often the model of choice. Ideal parameters are stored in the model database. The range information is processed in a prescribed manner to produce similar parameters, then the data and model parameters are matched. This an oversimplification of the process, but is sufficient to illustrate one of the main problems with this type of representation: a different model must be generated and stored for each individual object desired to be recognized. In our methodology, only one model is needed for each *class* of object.

Our goal is twofold: 1) to design a symbolic representation of three-dimensional object classes that can be used to represent generic objects in a variety of applications, and 2) to design and implement a matching procedure that can find correspondences between symbolic model features and 3D mesh data. To accomplish these goals, we develop a symbolic model for three-dimensional object classes and use knowledge of an object in a new, two-step, bottom-up/top-down approach to assist in matching a three-dimensional mesh representation of the object to the symbolic model. This information can be useful for many applications, some of which are:

1. More accurate and aesthetically pleasing reconstruction of an object.
2. Reconstruction from incomplete data.
3. Identification of features in medical images.
4. Recognition of objects in a limited environment.

Our new methodology uses symbolic models to represent 3-dimensional object classes. Symbolic representation of spatial information is not new. In [4], Hernandez presents an entire system based on qualitative spatial knowledge to describe 2D spatial objects and relationships. He proposes a way to extend his system to describe 3D objects based on volumetric primitives. He also provides a comprehensive survey of other work in the area of qualitative spatial representation. The approaches surveyed in this work have one thing in common; they use the qualitative representation for *reasoning* about spatial objects, but do not try to tie the qualitative model to any particular instance of an object. Other researchers have used qualitative spatial information for object recognition. Brooks' [2] ACRONYM system uses spatial relationships of volumetric models based on generalized cylinders to predict the occurrence of 2-D features in gray-tone images. Dickinson *et al.* [3] employ a qualitative shape model based on simple volumetric primitives called geons [1]. In the GRUFF system [10], Stark and Bowyer use symbolic models to represent categories of objects and use functional characteristics to decide if an object belongs to a particular category.

2. Working Model

Motivated by medical applications, we developed an abstract symbolic model to describe any three-dimensional object. The abstract model consists of a spatial object ontology and three networks: the *topology network*, the *part_of* network, and the *spatial relationships* network. This model is useful for reasoning about the object and can be the starting point for domain-specific models to be used in matching. The abstract model has been adapted to the domain of anatomy and has already demonstrated its usefulness in that domain. The Digital Anatomist Structural Abstraction [8], derived from this model, is an integral part of the Foundational Model of Anatomy [9].

The abstract symbolic model is useful for reasoning about the object, as discussed in [9], but it has no mechanism to identify itself with a real object. The *working model* provides that mechanism.

The working model is a structural model that consists of the set of features F of the object we desire to find and a set of global constraints R associated with those features. Each feature has five main properties: *label*, *working-set-types*, *shape-type*, *shape-parms* and *detectability*. The label is just the name of the feature. Working-set-types and shape-type are properties used by the algorithm that detects the feature. There may be any number of working sets associated with a feature and for each there will be a working-set-type. The working-set-type limits the sets of mesh points that may be associated with the feature. Shape-type limits the shape of the detected feature and has shape parameters, a fit-function, and an error function associated with it. Shape-parms is the set of shape parameter values for that feature. Detectability is the likelihood of finding this particular feature; it is used in the feature labeling algorithm.

3. Matching

Given a working model of a 3D object class and a 3D mesh data set representing a sensed instance of that class, *matching* is the process of finding the features of the model as subsets of the mesh points that satisfy the shape requirements and relationship requirements of the model. Our matching methodology has three stages: working set construction, feature construction, and feature labeling.

The first stage of the matching methodology is bottom-up, and its purpose is to find, for each feature of the working model, one or more candidate sets of points in the mesh where this feature may be found. The candidate sets of points are called *working sets*. More formally, a working set $W(f)$ of mesh M for feature f with working-set-type t is a subset $W(f) \subseteq P(M)$ of the point set $P(M)$ of the mesh M satisfying:

1. $W(f)$ is a connected point set.
2. t is a valid type for feature f.
3. $W(f)$ satisfies the requirements for type t.

Given a working model Q with feature set F, the first stage of matching is to identify for each feature $f \in F$ a set of working sets $S(f)$ for that feature. Thus the output of this stage is $S(F) = \{S(f) | f \in F\}$

The second stage is top-down, and its purpose is to further reduce the number of candidate sets of mesh points for each feature by ensuring the working sets still under consideration have the proper general shape and the parameters of that shape fall within the range of the shape parameters of that feature. These remaining working sets, in conjunction with the fitted shape-type of the feature, are called *constructed features*. In other words, a constructed feature $C(f)$ of mesh M for feature f with shape-type s and shape-parms m is a 3-tuple $C(f) = (WS(f), G(WS), E(WS))$, where $WS(f) \subseteq S(f)$ is either a single connected working set or the union of two or more working sets, $G(WS)$ is the set of parameters p resulting from a best-fit of s to $WS(f)$ and $E(WS)$ is a fit-error metric associated with $G(WS)$. $C(f)$ must satisfy the following conditions:

1. s is a valid shape-type for feature f.
2. Each parameter $p \in G(WS)$ falls within the given range of the corresponding parameter m in feature f.
3. $E(WS)$ must not exceed some given error threshold t_{err}.

Note that a constructed feature can be formed from more than one working set. This will allow generation of features that may be more robust than those generated from just one working set. For example, the fillets where the handle on a coffee cup connects to the body are usually detectable, but they are often small and can be mistaken for noise in the mesh. We use a constructed feature that consists of a line segment fit to *both* fillets, then base the constraints on properties of that line segment.

Given a working model Q with feature set F, the second stage of matching is to construct for each feature $f \in F$ a set of constructed features $T(f)$ for that feature. Therefore the output of the second stage is $T(F) = \{T(f) | f \in F\}$

The last stage of the matching methodology is also top-down, and its purpose is to find a consistent labeling from the working model features to the constructed features. Let $f \in F$ be a working model feature and $T(f)$ be its set of constructed features. Let R be the set of constrained relationships in the working model; $R = \{r_1, \ldots, r_m\}$, where each $r_i \in F^n \times \Upsilon$ is a labeled tuple of features $(f_1, \ldots, f_n, \rho)$ in which $f_i \in F$, and $\rho \in \Upsilon$ is the type of relationship. A *consistent labeling* $h : F \rightarrow T(F)$ is a function that maps model features to constructed features and must satisfy the following:

1. h is one-to-one, i.e., $h(f_i) = h(f_j) \Rightarrow f_i = f_j$
2. If a relationship between features is in R, then the parameters in the associated constructed feature must also satisfy the relationship.

There are many methods in the literature for solving consistent labeling problems; our current implementation is a backtracking tree search.

4. Implementation

To test our methodology, we implemented a prototype system using the working model described above. We developed four working set types based on curvature of the surface in a neighborhood. We implemented three functions for shape type, and allowed five types of constraint relationship between the features.

Working Set Types We used a numerical method, developed by Itter and Jain [5], to produce curvature estimates at each point p of a mesh. The working set types based on these local curvature measures are PLANAR (very low curvature), CONCAVE (high magnitude minimum curvature, CONVEX (high magnitude maximum curvature), and OTHER (curvature between PLANAR and CONCAVE/CONVEX). Curvature is calculated at each vertex and each is classified according to its type. Regions of connected, same-type vertices are grown, and small isolated regions merged into larger ones. The regions are then classified according to vertex type.

Shape Types We developed three allowable shape types that were based on the features present in the object classes we used in our experiments. The shape types are circle (used to describe circular features), line segment group (used to describe piecewise linear features), and plane (used to describe planar regions). Powell's method [6] of optimization is used to fit the shape type to the working set by minimizing an error function. The shape parameters in the working model are used to both provide initial estimates for the shape fitting to the working set, and to remove those working sets from consideration whose final shape fit does not meet the parameters.

Relationships The relationships we instantiated for our experiments are *parallel*, *perpendicular*, *reflective symmetry*, *centroid_distance*, and *concentric*. Features are parallel or perpendicular to one another if their best-fit planes are parallel or perpendicular. Two features have reflective symmetry if a reflective plane between them can be found. Centroid_distance places bounds (relative to the bounding box) on the distance between the centroids of two features. The concentric relationship is only valid for circular features. The planes of the circles must meet the *parallel* constraint, and the line segment formed by the centers of the circles must be perpendicular to the planes of the both circles.

5. Experiments and Results

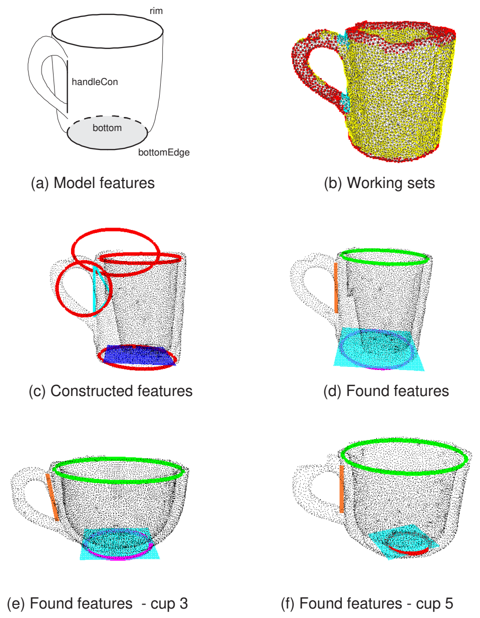
We implemented three working models for initial testing. The object classes of the models are pipe fitting, cup and airplane. We experimented with five instances of pipe fitting, five instances of cup, and three instances of airplane. For each instance of cup and pipe we used three different mesh resolutions. Details of all models and experimental results can be found in [7]; the cup class is discussed here.

In our working models for these objects, we used three of the four working set types: convex, concave, and planar, and all three shape types. The shape parameters allow us to specify ranges for the fitted shapes based on the bounding box of either the object or the working set(s).

The coffee cup model has four features: rim, bottomEdge, bottom, and handleCon. See Figure 1(a) for a sketch of the features. Rim, bottomEdge and bottom are self-explanatory, while handleCon is a feature constructed by drawing a line segment between the top and bottom handle fillets. There are four relationships defined between the features of the cup:

1. The rim and bottomEdge must be concentric.
2. The distance between the bottomEdge and the bottom must be less than one-fourth the smallest dimension of the bounding box of the object.
3. The handleCon must be perpendicular to the rim.
4. The handleCon must be perpendicular to the bottomEdge.

The experiments consist of two components: a qualitative part and a quantitative part. Mesh resolutions of twenty thousand, ten thousand, and five thousand faces were generated for each object of class "cup". We generated five instances of the coffee cup class. Each cup has different shape variations which help test our theory of a generic feature finder. Figure 1 shows the processing stages for one of the coffee cups. Part (b) shows the working sets obtained from the bottom up stage of processing. Constructed features are shown in part (c). Part (d) shows the features identified by our system. Figures 1(e) and (f) are examples of two other cup instances used in our experiments.



(a) Model features (b) Working sets

(c) Constructed features (d) Found features

(e) Found features - cup 3 (f) Found features - cup 5

Figure 1. Processing stages for cup 1 with a mesh resolution of 20K faces (a)-(d). Found features of two other instances of cup are shown in (e) and (f).

Qualitative Testing The qualitative part of our testing compared features present, features labeled correctly, features present but not labeled, and features labeled incorrectly to input from a human operator. The system was successful in 12 out of 14 tests of distinguishing between different types of pipe fittings based on relationships between features. The airplane was our most complex object, but the meshes obtained were of such poor quality that the working sets were not reliably grouped together and the results were inconclusive. Using the coffee cup model, our system reliably labeled the correct features 50 out of 60 times on five different cups with different shaped bodies and different types of handles (we had 3 resolutions of each cup instance). We found that the lowest resolution meshes used ($\sim$ 5K triangles) often did not provide enough detail for good separation of working sets.

Quantitative Testing The quantitative part of testing took a human operator's input as "ground truth" and calculated average distance from the constructed feature to the actual feature. The mesh was displayed and a human operator marked several vertices as belonging to a particular feature. Our system was then used to label the features, and the mean distance from the constructed feature to the operator-labeled vertices was calculated. Table 1 illustrates the distances for the cup objects.

Table 1. Distance calculations for coffee cups.

Mean Distance (mm)

	5K	10K	20K
Rim	3.92	2.99	2.56
Bottom Edge	6.92	2.86	1.93
Bottom Surf.	1.56	0.66	0.45
Handle	6.49	6.22	4.46

6. Conclusions

In this paper we have defined a 3D symbolic spatial model to represent generic object classes. We have presented a new methodology for detecting predefined features in object meshes, and a method for linking the symbolic model with a real instance of an object mesh. We have developed and implemented a prototype system using the model and methodology, and have shown the abstract model and the theory to be feasible approaches to the problem of recognizing and/or labeling the features of abstract object classes.

References

[1] I. Biederman. "Human Image Understanding: Recent Research and a Theory," *CVGIP*, Vol. 32, 1985, pp. 29-73.

[2] R. A. Brooks. "Symbolic Reasoning Among 3-D Models and 2-D Images," *Artificial Intelligence*, Vol. 17, Nos. 1-3, 1981, pp. 285-348.

[3] S. J. Dickinson, D. Metaxas and A. Pentland. "The Role of Model-Based Segmentation in the Recovery of Volumetric Parts from Range Data," *IEEE PAMI*, Vol. 19, No. 3, March 1997, pp. 259-267.

[4] D. Hernandez. Qualitative Representation of Spatial Knowledge, Springer-Verlag Berlin Heidelberg, Germany, 1994.

[5] D. J. Itter and A. K. Jain. "3-D Surface Discrimination from Local Curvature Measures," *Proc. CVPR '85*,IEEE Comput. Soc. Press, Silver Spring, MD, 1985, pp 119-23.

[6] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. Numerical Recipes in C, Cambridge University Press, New York, 1992.

[7] P. J. Neal. *Finding and Matching Topographic Features in 3-D Object Meshes.* PhD Thesis, University of Washington, 1999.

[8] P. J. Neal, L. G. Shapiro and C. Rosse. "The Digital Anatomist Structural Abstraction: a Scheme for the Spatial Description of Anatomical Entities", *Proc. 1998 AMIA Annual Symposium*,7-11 November 1998.

[9] C. Rosse, L. G. Shapiro and J. F. Brinkley. "The Digital Anatomist Foundational Model: Principals for Defining and Structuring its Concept Domain," *Proc. 1998 AMIA Annual Symposium*, 7-11 November 1998.

[10] L. Stark and K. W. Bowyer. *Generic Object Recognition Using Form and Function.* New Jersey. World Scientific Publishing, 1996.