Sticks, Plates, and Blobs: A Three-Dimensional Object Representation for Scene Analysis

Linda G. Shapiro
Prasanna G. Mulgaonkar

John D. Moriarty
Robert M. Haralick

Virginia Polytechnic Institute and State University
Department of Computer Science

ABSTRACT

In this paper, we describe a relational modeling technique which categorizes three-dimensional objects at a gross level. These models may then be used to classify and recognize two dimensional views of the object, in a scene analysis system.

I. Introduction

The recognition of three-dimensional objects from two-dimensional views is an important and still largely unsolved problem in scene analysis. This problem would be difficult even if the two-dimensional data were perfect, but the data can be noisy, distorted, occluded, shadowed and poorly segmented, making recognition much harder. Since the data is so rough it seems reasonable that very rough models of three-dimensional objects should be used in the process of trying to classify such data. In this paper we describe a relational model and discuss its use in a scene analysis system.

There have been many approaches to modeling three-dimensional objects. For a comprehensive collection see the proceedings of the Workshop on Representation of Three-Dimensional Objects [13]. Also see Voelcker and Requicha [11] and Brown [4] for mechanical design; York et.al. [14] for curved surface modeling using the Coons surface patch [5]; Horn [6] and Waltz [12] for study of light and shadows; Badler et.al. [2] for study of human body modeling; and Agin and Binford [1] and Nevatia and Binford [7] for the generalized cylinder approach. The models we suggest are related to the generalized cylinder models, but are rougher descriptions that specify less detail about three-dimensional shape than do generalized cylinders.

II. Sticks, Plates, and Blobs in Relational Descriptions

A relational description of an object consists of a set of parts of the object, the attributes of the parts, and a set of relations that describe how the parts fit together. Our models have three kinds of three-dimensional parts: sticks, plates, and blobs. Sticks are long, thin parts having only one significant dimension. Plates are flattish wide parts consisting of two nearly flat surfaces connected by a thin edge between them. Plates have two significant dimensions. Blobs are neither thin nor flat; they have three significant dimensions. All three kinds of parts are "near convex"; that is a stick cannot bend very much, the surfaces of a plate cannot fold too much, and a blob can be bumpy, but cannot have large concavities. Figure 1 shows several examples of sticks, plates, and blobs.

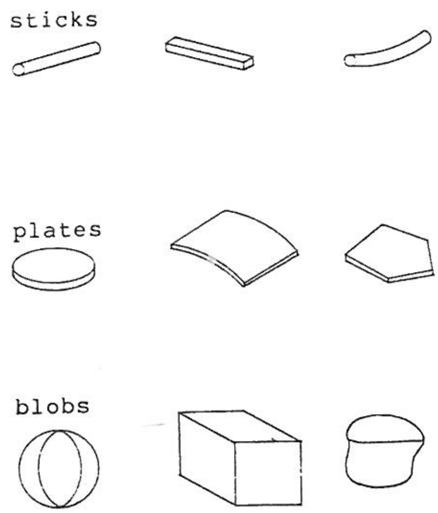

Figure 1 illustrates several examples each of sticks, plates and blobs.

In describing an object, we must list the parts, their types (stick, plate, or blob), and their relative sizes; and we must specify how the parts fit together. For any two primitive parts that connect, we specify the type of connection and up to three angle constraints. The type of connection can be end-end, end-interior, end-center, end-edge, interior-center, or center-center where "end" refers to an end of a stick, "interior" refers to the interior of a stick or surface of a plate or blob, "edge" refers to the edge of a plate, and "center" refers to the center of mass of any part.

This research was supported by the National Science Foundation under grants MCS-7923827 and MCS-7919741.

For each type of pairwise connection, there are one, two, or three angles that, when specified as single values, completely describe the connection. For example, for a stick and a plate in the end-edge type connection, two angles are required: the angle between the stick and its projection on the plane of the plate and the angle between that projection and the line from the connection point to the center of mass of the plate.

Requiring exact angles is not in the spirit of our rough models. Instead we will specify permissible ranges for each required angle. In our relational model, binary connections are described in the CONNECTS/SUPPORTS relation which contains 10-tuples of the form (Part1, Part2, SUPPORTS, HOW, VL1, VH1, VL2, VH2, VL3, VH3) where Part1 connects to Part2, SUPPORTS is true if Part1 supports Part2, HOW gives the connection type, VLi gives the low-value in the permissible range of angle i and VHi gives the high value in the permissible range of angle i, i = 1, 2, 3.

The CONNECTS/SUPPORTS relation is not sufficient to describe a three-dimensional object. One shortcoming is its failure to place any global constraints on the resulting object. We can make the model more powerful merely by considering triples of parts (s1,s2,s3) where s1 and s3 both touch s2 and describing the spatial relationship between s1 and s3 with respect to s2. Such a description appears in the TRIPLE CONSTRAINT relation and has two components: 1) a boolean which is true if s1 and s3 meet s2 on the same end (or surface) and 2) a contraint on the angle subtended by the center of mass of s1 and s3 at the center of mass of s2. The angle constraint is also in the form of a range.

Our current relational description for an object consists of ten relations. The A/V relation or attribute-value table contains global properties of the object. Our A/V relations currently contain the following attributes: 1) number of base supports, 2) type of topmost part, 3) number of sticks, 4) number of plates, 5) number of blobs, 6) number of upright parts, 7) number of horizontal parts, 8) number of slanted parts. The A/V relation is a simple numeric vector, including none of the structural information in the other relations. It will be used as a screening relation in matching; if two objects have very different A/V relations, there is no point in comparing the structure-describing relations. We are also using the A/V relations as feature vectors to input to a clustering algorithm. The resulting clusters represent groups of objects which are similar. Matching can then be performed on cluster centroids instead of on the entire database of models. Other relations include SIMPLE PARTS, PARALLEL PAIRS, PERPENDICULAR PAIRS, LENGTH CONSTRAINT, BINARY ANGLE CONSTRAINT, AREA CONSTRAINT, VOLUME CONSTRAINT, TRIPLE CONSTRAINT and CONNECTS/SUPPORTS.

III. Matching

Relational matching of two-dimensional objects to two-dimensional models is a well-defined operation. See Barrow, Ambler, and Burstall [3] for a discussion of exact relational matching, Shapiro [8] for relational shape matching, and Shapiro and Haralick [10] for inexact matching. Our problem in scene analysis is to match two-dimensional perspective projections of objects (as found in an image) to the three-dimensional models stored in the database. Our approach to this problem is to analyze a single two-dimensional view of an object, produce a two-dimensional structural shape description, use the two-dimensional description to infer as much as possible about the corresponding three-dimensional description, and then use inexact matching techniques in trying to match incomplete and possibly erroneous three-dimensional object descriptions to our stored three-dimensional relational models.

We decompose a two-dimensional view into simple parts by a graph-theoretic clustering scheme as described in [9]. To match a two-dimensional object description to a three-dimensional model is to find a mapping from the two-dimensional simple parts of the object to the sticks, plates and blobs of the model so that the relationships among the two-dimensional parts are not inconsistent with the relationships among the three-dimensional parts. For example, a binary CONNECTS relation can be constructed for the two-dimensional parts. For a pair (p1,p2) of three-dimensional model parts where (p1,p2,*,*,*,*,*,*,*,*,) is an element of the CONNECTS/SUPPORTS relation and a mapping h from three-dimensional model parts to two-dimensional object parts, if (h(p1),h(p2)) is not an element of the two-dimensional CONNECTS relation, then an error has occured. If a mapping accumulates too many errors from various n-tuples of various relations not being satisfied, that mapping cannot be considered a match.

As an example, suppose the three-dimensional model of a simple chair contains two plates (the back B and seat S) and four sticks (legs L1, L2, L3, L4). The relation obtained from just the first two columns of the CONNECTS supports relation is {(S,B), (B,S), (L1,S), (S,L1), (L2,S), (S,L2), (L3,S), (S,L3) (L4,S), (S,L4)}. Now consider the two-dimensional decomposition of Figure 2. We can construct the hypothetical connection relation C = {(s1,s2), (s2,s1), (s3,s2), (s2,s3), (s3,s1), (s1,s3), (s4,s2), (s2,s4), (s4,s1), (s1,s4), (s5,s2), (s2,s5)}. Then the mapping f defined by {(S,s2), (B,s1), (L1,s3), (L2,s4), (L3,s5), (L4,s4)} accumulates no error while the mapping g defined by {(S,s1),(B,s2), (L1,s3), (L2,s4), (L3,s5), (L4,s4)} accumulates error since (L3,S) is in the model, but (f(L3),f(S)) = (s5,s1) is not in C.

Not all of the three-dimensional relations can be directly constructed from two-dimensional data. (If they could, the entire scene analysis problem would be much easier.) For example, only an estimate of whether one part supports another can be computed. Relations like PARRALLEL PAIRS and LENGTH CONSTRAINT can also be estimated. Relations involving angles are probably the most dif-

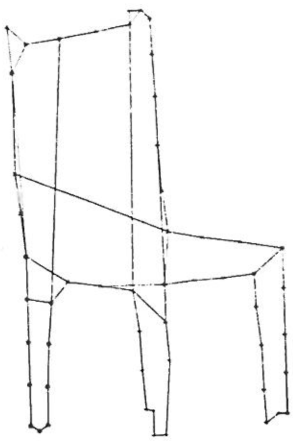

Figure 2 illustrates the decomposition of a two-dimensional chair by graph-theoretic clustering.

ficult, since a perspective projection will change the angles between parts. Such information should be left out of initial matching attempts and used later to try to validate a given match or to choose between several possible matches. The precise definition of an inexact match from a two-dimensional description to a three-dimensional description is the subject of our current research.

IV. Summary of Current and Future Research

We have described a relational model for three-dimensional objects, in which the parts of an object are very grossly described as sticks, plates, or blobs. We are building a database of three-dimensional object models. The objects in the database are being clusterd into groups, using graph-theoretic clustering algorithm. Instead of comparing a two-dimensional view to every object in the database, it will be compared initially only to a centroid objects in each group. Only in those groups where the unknown object is most highly related to the centroid will any full relational matching take place.

Relational matching will be a form of the inexact matching we described in [10]. The general method will be to obtain estimates of the three-dimensional relations from the two-dimensional shape and match these estimates against the three-dimensional models. Deriving the algorithms and heuristics for the matching is one of our most challenging tasks.

References

1. Agin, G.J. and T.O. Binford, "Computer Descriptions of Curved Objects", IEEE Transactins on Computers, Vol. 25, No. 4, April 1976.
2. Badler, N.I., J. O'Rourke, and H. Toltzis, "A Spherical Representation of A Human Body For Visualizing Movement", Proceedings of the IEEE, Oct. 79.
3. Barrow, H.G., A.P. Ambler, and R.M. Burstall, "Some Techniques for Recognizing Structure in Pictures", Frontiers of Pattern Recognition, S. Watanabe (ed), Academic Press, New York, 1972, pp. 1-29.
4. Brown, C.M., A.A.G. Requicha and H.B. Voelcker, "Geometric Modeling Systems for Mechanical Design and Manufacturing", Proc. ACM 1978, Washington D.C., Dec 4-6, 1978.
5. Coons, S.A., "Surfaces for Computer-Aided Design of Space Forms," M.I.T. Project MAC, MAC-TR-41, June 1967 (AD 663504).
6. Horn, B., "Obtaining Shape From Shading Information", in The Psychology of Computer Vision, (P.H. Winston, ed.), McGraw-Hill, New York, 1975, pp. 115-155.
7. Nevatia, R.K. and T.O. Binford, "Structured Descriptions of Complex Objects", Proc. Third International Joint Conference on Artificial Intelligence, Stanford, 1973.
8. Shapiro, L.G., "A Structural Model of Shape", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No. 2, March 1980.
9. Shapiro, L.G. and R.M. Haralick, "Decomposition of Two-Dimensional Shapes by Graph-Theoretic Clustering," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-1 pp. 10-20, Jan. 1979.
10. Shapiro, L.G. and R.M. Haralick, "Structural Descriptions and Inexact Matching", Technical Report CS79011-R, Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, Nov. 1979.
11. Voelcker, H.B. and A.A.G. Requicha, "Geometric Modeling of Mechanical Parts and Processes", Computer, Vol. 10, No. 12, Dec. 1977, pp.48-57.
12. Waltz, D., "Understanding Line Drawings of Scenes with Shadows", in The Psychology of Computer Vision, (P.H. Winston, ed.), McGraw-Hill, New York, 1975.
13. Workshop on Representation of Three-Dimensional Objects, (R. Bajcsy, Director), University of Pennsylvania, May 1-2, 1979.
14. York, B., A. Hanson, and E.M. Riseman, "Representation of Three-Dimensional Objects with Coons Patches and Cubic B-Splines", Department of Computer and Information Science, University of Massachusetts, Amherst, MA 01003.