

3D OBJECT RE-PRESENTATION AND MATCHING WITH B-SPLINES AND SURFACE PATCHES

Bryant W. York
IBM Research Laboratory, San Jose, Ca

Allen R. Hanson
Edward M. Riseman
University of Massachusetts, Amherst, Ma

ABSTRACT

An approach to 3D object representation and matching, which employs cubic B-spline curves and Coons surface patches is described. 3D objects are represented as structured collections of surface patches whose boundary curves are approximated by cubic B-splines. Both curved and polyhedral objects may be represented. A variety of 3D and 2D shape features are defined which are easily computed for this representation. Many of the 3D features have 2D analogues, thus providing access paths from 2D to 3D. It is anticipated that this approach will prove useful for matching 3D object models against 2D object descriptions. The representation, our approaches to the problems of indexing and matching, the state of the implementation, and our experiences with the implementation are briefly discussed.

1. 3D OBJECT REPRESENTATION

3D Objects are represented by structured collections of surface patches. A surface patch may be simple or complex. Simple patches are represented in Coons form [1] with boundary curves, across boundary derivative curves, and blending functions represented as cubic B-spline curves. A complex surface patch is formed by joining two or more patches along a common boundary. For more details on the representation see [8,9].

Two of the more significant properties of the representation are the ability to represent both curved and polyhedral objects in a single representation and the large number of shape features which it admits. A cubic B-spline curve is represented by a list of points (x-y-z coordinates) called the "vertex polygon." The multiple-vertex property of cubic B-splines permits slope-discontinuous and curvature-discontinuous curves to have similar representations to smooth curves; the only difference is that a discontinuous curve will have at least one vertex of multiplicity greater than 1 in its vertex polygon. A circle may be approximated by a cubic B-spline with only 4 vertices in its vertex polygon and a square is approximated by a similar vertex polygon with 4 vertices of multiplicity 3 (see Figure 1). When cubic B-splines are used to represent the boundary curves of a Coons surface patch, both curved and polyhedral objects may be obtained. Figure 2 shows a plot without hidden line removal of a smooth, curved object (a running shoe) described by a single patch in our representation; Figure 3 shows a polyhedral object (a wedge). Because of the smooth join property of Coons surface patches, complex smooth objects may be formed by joining several component patches, as well as objects with both curved and polyhedral parts. The vertex polygon of a cubic B-spline curve permits computation of polygonal shape features and the Coons patch form allows the computation of additional shape features which are characteristic of the patch rather than its boundary curves. Figure 4 lists some of the more significant shape features for B-spline curves and Coons patches.

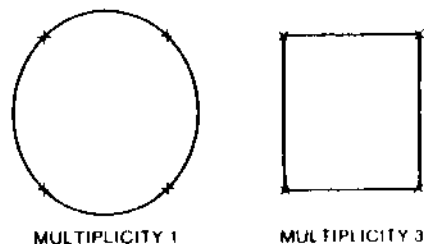


Figure 1: Circle and Rectangle

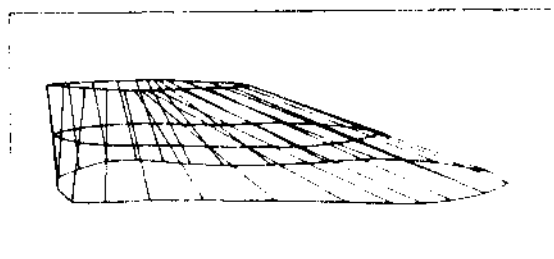


Figure 2: Running Shoe

We have interactively designed 2D and 3D cubic B-spline curves, stored them in a data base, computed their feature values, and associated them with surface patch definitions in any of three roles -- boundary curve, across boundary derivative curve, blending function. The design of Coons patches is less interactive. Simple Coons patches are designed by entering the coordinates of the corner position matrix and pointers to desired boundary curves and blending functions. This is a cumbersome process and it requires that the surface patch be designed on paper before it may be entered into the data base. In progress is the design of a mechanism for interactively constructing arbitrarily complex patches using a quad tree data structure and the Coons smooth join criteria. Such a mechanism will facilitate the construction of new object models from existing ones.

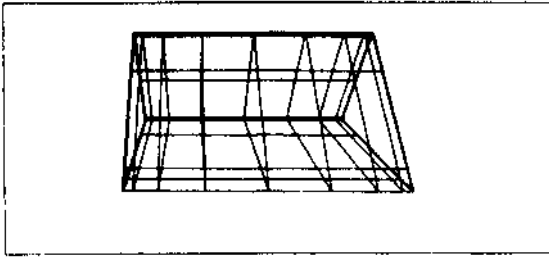


Figure 3: Wedge

Curves	
OC	= open/closed
PI	= planarity
SM	= smoothness
NV	= no. of vertices
MV	= no. of multi-vertices
C	= compactness
AV3D	= angular variability
MAXA1	= max angular turn
Patches	
C	= patch compactness
PVA	= position vector angles
TVA	= tangent vector angles
PVMR	= position vector magnitude ratios
TVMR	= tangent vector magnitude ratios

Figure 4 Some Shape Features

H ACCESSIBILITY

Accessibility deals with the problem of gaining access to a proper set of 3D object representations based upon information extracted from a 2D image. It involves the representation of 3D shape information, the representation of 2D shape information, and the indexing paths available in the data base.

A Structure of Visual Knowledge

In the VISIONS system [3] visual knowledge is collected in a layered network data base called long term memory (LTM). A high level of the network contains a single type of entity. The entity types used for this work are *schema*, *object*, *surface patch*, *3D curve*, *region*, *2D curve*, *vertex*. Each level is currently partitioned into four spaces where a space is merely a collection of entities. The four spaces are the "class" space, the "class feature" space, the "instance" space, and the "instance feature" space. An example of a class entity at the object level is, say, "table". This entity is a prototype from which many instances may be constructed. The instances may differ in a variety of ways. Consequently, separate feature spaces are maintained for classes and instances. Figure 5 is an example of a portion of the network structure of LTM. The advantage of this network structure is that it provides a straightforward, logical view of the current structure of stored visual knowledge and the semantics of access to particular knowledge is easily described in terms of path traversals [6].

B. 2D Representation

In the world of 2D scene analysis it is necessary to extract representations of 2D entities from digitized images. Our approach is to attempt to keep the 2D representation as close to the 3D representation as possible. Thus, the 2D digital curves which result from the segmentation of a digitized image are converted to cubic B-spline representation. A B-spline curve is fitted to an image curve by analyzing the K-curvature [2] of the image curve and placing knots along the digital curve at appropriate points. Special algorithms have been developed for knot placement and knot coalescing (the creation of multiple knots). The fitting procedure is iterative. First knots are placed at points of high K-curvature, then knots are moved or coalesced according to a set of special heuristics, the cubic B-spline through the knots is computed, and a measure of error of fit is computed. If the error is above some threshold, we iterate from the knot motion/coalescing step. When the error is below threshold, the knot vector is converted to the corresponding vertex polygon and 2D shape features are computed. The special heuristics are based upon smoothness criteria such as the detection of corners or peaks (slope discontinuities) and the detection of "staircase" straight lines. For more details see [9].

The algorithms for knot placement, knot motion, knot coalescing, B-spline computation, and error computation have been implemented. Currently the fitting procedure is interactively controlled with the user indicating the locations in the curve where a given special heuristic should be applied. We would eventually like to develop a mechanism which has good performance without user intervention.

C Indexing

A small LTM has been constructed containing 1K single patch objects. The graph representing this LTM contains approximately 150 nodes and 3000 edges. An indexing mechanism has been implemented in the (iRASPLR) [5] graph processing language. The mechanism provides for execution of relational searches in the network. An example of a relational search might be the query "Find all curves which are closed and have angular variability = 1.28". The current mechanism merely executes raw primitive queries on the 3D levels of ETM. ANDing and ORing of the result sets is done interactively when all of the primitive results are available.

We have not, as yet, implemented "standard views" [7]. Correlation of certain 2D shape features (such as AV_{1,1}) for standard views with the corresponding 3D features (AV_{1,0}) will provide indexing paths from the 2D levels to the 3D levels. Figures 6 and 7 demonstrate the importance of standard views and the change in 2D shape features with change in view. Four projections of a cubic B-spline approximation to a 3D circle and its associated vertex polygon are shown. The circle is defined in 3-space even though it lies in a plane. The plane of the circle stalls out orthogonal to the line of sight and is rotated away from the viewer by the angular amounts indicated. As the circle is rotated, it projects as an ellipse of greater and greater eccentricity until finally it projects as a straight line. Figure 8 shows certain 2D feature-values for rotations from 0° to 90°. In the case of the circle some of the 2D feature-values for () are equivalent to the 3D feature-values for the 3D circle. From such a table we can begin to correlate change in feature values with change in viewpoint. Ultimately we would like to develop correlations for more complex shapes.

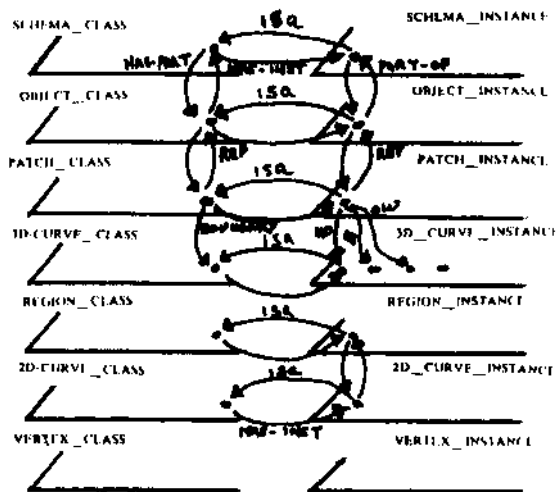


Figure 5: LTM Network Structure

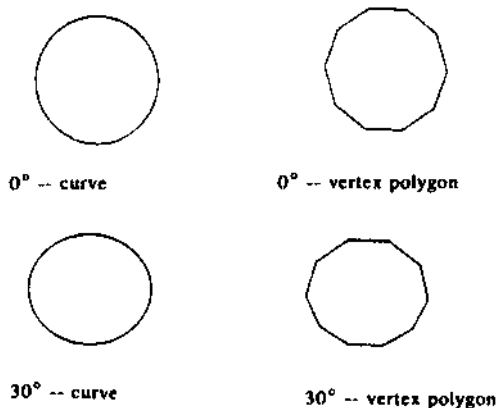


Figure 6: Projections of Circle -- 0° and 30°

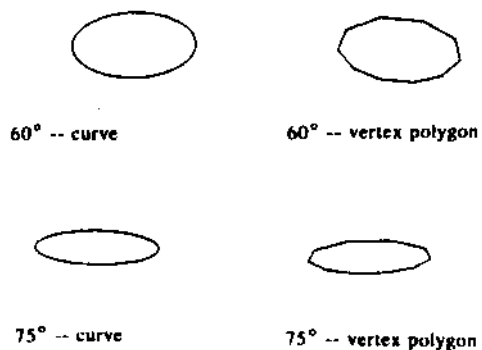


Figure 7: Projections of Circle -- 60° and 75°

	0°	30°	45°	60°	75°	90°
OC	1	1	1	1	1	0
PL	0	0	0	0	0	0
SM	0	0	0	0	0	0
NV	10	10	10	10	10	2
MV	0	0	0	0	0	2
VF	10	10	10	10	10	0
MV/NV	0	0	0	0	0	1
NT/NV	1	1	1	1	1	1
PGL/BL	0	0	0	0	0	1
C	12.57	12.61	12.91	14.62	21.11	0
MAX1(DEG)	144	150.6	156.2	163.2	171.3	0
MAX1/SIN1	1	1.28	1.55	2.16	3.91	1
AV _{2D}	0	.57	1.01	1.28	1.01	0

Figure 8: Shape Features for Projections of circle

111 CONCLUSION

Matching is a difficult problem involving exponential search. It becomes more tractable when the number of candidates against which a description must be matched is reduced. We feel that our representation, the structure of LTM, the use of standard views and multiple structural decompositions [4], and a powerful indexing mechanism will contribute greatly to this reduction.

REFERENCES

- [1] Coons, S. A. "Surfaces for computer-aided design of space forms. Department of Computer Science", M.I.T. Cambridge, Massachusetts. NTIS AD-663 504. 1967
- [2] Davis, I. S. and Rosenfeld, A. "Applications of relaxation labelling. 2 spring-loaded template matching". Technical report 440. Computer Science Center, University of Maryland, College Park, Maryland, 1976
- [3] Hanson, A. and Riseman, E. "VISIONS: a computer system for interpreting scenes". In *Computer Vision Systems*. A. Hanson and E. Riseman (eds) p. 303-334. Academic Press, New York, 1978
- [4] Hinton, G. 1979. "Some demonstrations of the effects of structural descriptions in mental imagery." *Cognitive Science* 3:3 (1979) 231-2M
- [5] Lowrance, J. D. "Grasper 1.0 Reference Manual". COINS technical report 78-20. Dept. of Computer and Information Sciences. University of Massachusetts. Amherst, Ma. 1978.
- [6] Marr, D. and Nishihara, H. "Representation and recognition of the spatial organization of three dimensional shapes". AI-Memo 377. MIT Artificial Intelligence Laboratory, MIT. Cambridge, Ma. August 1976

- [7] Minsky, M "A framework for representing knowledge" In The Psychology of Computer Vision. Ed P H Winston, McGraw-Hill. New York, 1975
- [8] York, B W., Hanson, A. R . and Riseman, I M "A Surface Representation for Computer Vision." In Proc. of IEEE Workshop on Picture Data Description Management Asilomar, Ca. August 1980, 124- 129
- [9] York, B W Shape Representation in Computer Vision. Ph.D Dissertation. University of Massachusetts. Amherst. Ma., May 1981