[24] SMS: A Suggestive Modeling System for Object Recognition

Robert B Fisher

Department of Artificial Intelligence
University of Edinburgh, Edinburgh EH1 2QL, UK

Reprinted, with permission of Butterworth Scientific Ltd, from *Image and Vision Computing*, 1987, **5**, 98-104.

1. Introduction

SMS [4,6] is an object representation system motivated by the requirements of recognition instead of depiction. It is designed for model invocation, reference frame estimation and matching (roughly as in IMAGINE [5], only with many extensions). It represents strongly visible features and relationships of non-polyhedral man-made objects, integrating curve, surface and volumetric structural descriptions in a subcomponent hierarchy.

The central principles of the modeling philosophy are:

- The models are suggestive rather than literal. By suggestive, we mean that: (1) surface shapes and volumes may be only approximate, (2) only salient features will be modeled, omitting minor or hard-to-segment features and (3) the model may not be completely closed or connected (e.g. representative surface patches may be used instead of complete surfaces). Literal models are suitable for image generation; suggestive models represent salient features without excessive metrical detail. Suggestiveness is needed for generic model representation, otherwise rough matchability is not possible.
- The modeled features should be observable data features. This facilitates matching without having to compute the visible appearance of a feature (Marr's accessibility criterion [12]).
- Both data and model features are assumed to be segmented similarly for matching correspondence and are symbolically described for efficiency.
- Marr's uniqueness criteria is to be slightly relaxed so there may not be a canonical description. Alternative object representations are allowed to cope with both incomplete descriptions and scale-based description change.
- Marr's scope, stability and sensitivity criteria [12] still apply.

It is presumed that most of the information in the model will be explicit, instead of being computed when necessary. This cannot always be the case because: (1) descriptions of incompletely constrained objects (e.g. variable size or flexibly connected) cannot be fully predictable and (2) the many less significant features create a combinatorial explosion in *a priori* description prediction, whereas their visibility is directly deducible given a roughly oriented model.

2. Requirements on SMS

Object representations are required for the following purposes:

- object feature and relationship descriptions are needed to constrain model-to-data matches and 3D location,
- visible object features and their configurations are needed for model invocation,
- predicted feature relationships are also needed to understand feature visibility from a given viewpoint.

The most important representation is the geometric model. From this, one can predict features and relationships as seen from any particular viewpoint, as well as verify observed relationships.

For matching, the geometric model should:

- represent strong edges,
- make surface information explicit, because surfaces are the primary visible features,
- make volumetric information explicit, because volumes represent the spatial distribution of the object, and because volumetric relationships can be deduced from the data when matchable surface information is not,
- be able to represent solid and laminar objects,
- have three dimensional, transformable representations for understanding appearance from arbitrary viewpoints,
- have geometric part-whole relationships, and
- allow partially constrained size and placement relationships.

Competent vision systems with large model bases need some form of model invocation. There are many approaches to this problem, but here it is based on accumulating evidence for objects, mediated by the associations between objects [5].

Direct evidence is computed by comparing the degree to which observed data properties meet modeled (unary or binary) property constraints, which must therefore be part of the model. Unary constraints specify the value ranges that are acceptable for different attributes of individual features. Binary constraints specify the 3D spatial configuration of the features. (Examples are the expected area of a surface and the angles at which two surfaces meet.) This information is made explicit for efficient invocation and matching.

The plausibility of related objects provides indirect evidence for the object, through the subcomponent, supercomponent, subtype (specialization) and supertype (generalization) relationships. Component relationships are implicit in the subcomponent hierarchy of the geometric model and the generic relationships must be given separately.

Invocation also requires subcomponent visibility groups, to indicate which possibly related objects contribute evidence for a given object. Each group specifies the major object features seen together from a given range of viewpoints. Only the prominent features and configurations are represented and only for significant viewpoint ranges.

3. Relationship to Previous Modeling Systems

The SMS models are most closely related to those used in ACRONYM [2] and IMAGINE [5]. The hierarchical reference frame and volumetric method used in SMS follows ACRONYM, though the primitive solids used are not generalized cylinders. IMAGINE used surfaces as its primitives in a subcomponent hierarchy to make explicit the shape of individual objects.

The volumetric primitives of Shapiro et. al. [15] were chosen to represent the essential character and relationships of solids.

Several recent 3D vision systems are based primarily on surface patch representations. Faugeras and Hebert [3] used an empirically derived fragmentary planar patch decomposition of an irregular object's surface, (patches were characterized by nominal position and surface orientation). Grimson and Lozano-Perez [8] used a similar representation. The models also included additional information, such as angles between normals and distances between points to improve recognition efficiency.

Bolles et al [1] used a vertex, edge, surface and volume representation, linked in a winged-edge-like representation. Features were also represented in classification (size,type) trees, to promote quick indexing of candidate models from observed data.

Many modeling systems use wire frames, and while no complete wire frame is used here, strong object edges are represented.

The variable and constraint method of ACRONYM has been followed with some modifications. The specialization method is similar to that of Marr and Nishihara [13], where specializations have different structural models and are linked by subcomponent and generic indices to associated models.

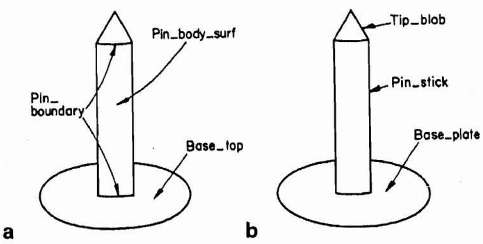

Figure 1: Drawing Pin Model

The viewer-centered representation is based on the subcomponent group of IMAGINE, the view potential of Koenderink and van Doorn [10] and the aspect graph information proposed for the YASA representation [7]. Here, descriptions of structures are represented according to their visibility and apparent configuration from given viewpoints.

The terms for local relations between solids follow Shapiro et al [15]. The axis relations are similar to those given by Marr [12] and express the relative size and placement of axes.

4. A Brief Summary of SMS

This section illustrates the contents of an SMS model (figure 1) through use of a drawing pin model.

SMS's data primitives are viewpoint independent object features organized in an object-centered geometric reference frame. The primitives are chosen for their visible salience:

space curves - represented by curvature and extent. Closed ellipses are represented explicitly, and other space curves are assumed to be segmented into straight lines and circular arcs. (Curves with torsion are not modeled, but this would be an easy extension.)

surfaces - segmented by roughly constant principal curvatures, and represented using surface patches from a torus (because its two surface curvatures correspond with the two observable principal curvatures). Degenerate and other cases such as planes, cylinders and cones are also represented. Patch boundaries are nominal and are defined using space curves.

volumes - represent extended spatial distributions, and have primarily 1, 2 or 3 directions of extension. The three first-order primitives are the STICK, PLATE and BLOB, which roughly characterize mass distribution without precise surface shape description. The extensions are parameterized, so slightly distorted volumes are allowed. Recently, second order volumetric primitives have been added to improve model sensitivity (see section 4.1).

These primitives and their parameters are used because they correspond closely with descriptions derived from 2 1/2 D sketch data. The different feature types are treated as alternatives, because of data unpredictability. Hence, a model will contain a mixture of each of the three types, and the intention is that evidence of any type would be sufficient.

Examples of feature definitions in the pin model are:

space curve - the curve of the orientation discontinuity where the pin shaft meets the base.

```
(ELLIPSE pin_boundary
    MAJOR_RADIUS    0.1
    MINOR_RADIUS    0.1
)
```

surface patch - the spherical patch of the top surface of the pin base. The negative minor radius declares the surface to be concave. The definition defines a complete torus, which needs to be trimmed by patch boundaries to form the object surface. The boundary list shows several curves that lie approximately on the surface, including the pin_boundary defined above. These boundaries delimit a spherical cap with a hole in it; the included point designates which region of the segmented torus is the patch. Translations and rotations are described below. The scale factor allows local rescaling of features.

```
(TORUS base_top
  MAJOR_RADIUS 0.0
  MINOR_RADIUS -1.0
  BOUNDARY_LIST   (

    ((PLACED_FEATURE base_boundary
      AT TRANSLATION (0,0,0.84)
        ROTATION VECTOR
          (0,0,-1) (0,0,-1)
      SCALE 1.0
    ))

    ((PLACED_FEATURE pin_boundary
      AT TRANSLATION (0,0,0.995)
        ROTATION VECTOR
          (0,0,-1) (0,0,-1)
      SCALE 1.0
    )))

  (INCLUDED_POINT (0.2,0,0.98))
)
```

volume - a bent PLATE is the volumetric approximation to the pin head, having two directions of extension. The thickness of the plate is small relative to the radius

```
(PLATE base_plate
  RADIUS      0.58
  THICKNESS     0.1
  BEND        0.85
)
```

Assemblies are formed from previously defined subassemblies and surfaces, where the features are placed using reference-frame transformations. The assemblies also record volumetric relationships between the solids, such as whether a STICK connects to a PLATE in the center or the edge, and relative relationships between volume axes, such as size, orientation and placement.

Reference frame rotations are specified in three forms, according to whether:

(1) the rotation is completely constrained (but may be a variable quantity, as in a robot joint angle),

(2) the rotation is symmetric about an axis, or

(3) the rotation is completely unconstrained, as with a spherically symmetric feature.

Translations are specified by a transforming vector (possibly variable, too).

Two types of assembly are defined. The first is a PRIMARY_ASSEMBLY, whose role is to group alternate representations (e.g. curves, surfaces or volumes) for primitive unstructured objects, because alternative evidence may be available for the recognition of primitive features. Any evidence should be allowed, without making the existence of the structure in larger structures contingent on the type of data evidence.

The PRIMARY_ASSEMBLY for the cylindrical pin is given below. Here, the curve, surface and volume alternatives are placed in the reference frame for the whole assembly. The ASM_ALT sections separate the equivalent alternative evidence groups. The PLACED_FEATURE blocks place an instance of the named feature in the object's local coordinate frame. The AT block gives the reference frame transformation from the feature's local frame to that of the object, with the required TRANSLATION and ROTATION.

```
(PRIMARY_ASSEMBLY pin_body

  (VARS (NONE))

  ((ASM_ALT /* curves */

    ((PLACED_FEATURE pin_boundary
      AT TRANSLATION (0,0,0)
        ROTATION VECTOR
          (0,0,-1) (-1,0,0)
      SCALE 1.0)

    (PLACED_FEATURE pin_boundary
      AT TRANSLATION (length,0,0)
        ROTATION VECTOR
          (0,0,-1) (1,0,0)
      SCALE 1.0)))

  (ASM_ALT /* surfaces */
    ((PLACED_FEATURE pin_body_surf
      AT TRANSLATION (0,0,0)
        ROTATION VECTOR
          (1,0,0) (1,0,0)
      SCALE 1.0)))
```

```
(ASM_ALT /* volumes */
  ((PLACED_FEATURE pin_stick
      AT TRANSLATION ((length/2),0,0)
        ROTATION VECTOR
          (1,0,0) (1,0,0)
      SCALE 1.0))))

 ( /* structure properties */ NONE)
)
```

The second type of assembly is the STRUCTURED_ASSEMBLY, whose role is to group subcomponents into an object, using the reference frame transformation mechanism. An example of this is shown here, where the pin_head and pin_body PRIMARY_ASSEMBLYs are joined to form the pin assembly. The assembly also has new properties specified. The first constraint states that the head and body surfaces are adjacent. Full path names are used because the referenced features are not always defined at the current level of assembly. The connection constraints describe the relationship that the volumetric primitives have (following [15]). Here, the END of the pin_body, a STICK, is attached to the CENTER of the pin_head, a BLOB.

```
(STRUCTURED_ASSEMBLY pin

   (VARS (length (DEFAULT_VALUE 1.0)))

   (/* substructures */
    (PLACED_FEATURE pin_body
      AT TRANSLATION (0,0,0)
        ROTATION VECTOR
            (1,0,0) (1,0,0)
      SCALE 1.0)

    (PLACED_FEATURE pin_head
      AT TRANSLATION (length+0.2,0,0)
        ROTATION VECTOR
            (1,0,0) (-1,0,0)
      SCALE 1.0))

    ( /* properties */
      (CONNECTED
          pin_head->pin_head_surf
           pin_body->pin_body_surf)
      (CONN_CONST pin_body
           END_CENTER pin_head))
)
```

Variables represent incompletely determined aspects of the models, such as shape, size or relative position and are bound in local contexts. Use of variables follows structured programming techniques and define the contexts within which variables are bound (dynamic binding). The defining context is the smallest hierarchical superobject context binding the variable. A robot finger joint angle can then be defined in the context of the finger only, so has a distinct value for each finger instance. If a hand scale variable is then defined in the context of the hand, but referenced in each finger subcontext, it has a distinct value in each hand instance and the same value in each finger subinstance.

Constraints on expressions containing variables are allowed, as in ACRONYM. The following limits the value of the "length" variable in the "pin" context:

```
(CONSTRAINT ((length > 0.5)) ASSEMBLY pin)
(CONSTRAINT ((length < 2.0)) ASSEMBLY pin)
```

There is a hierarchy of descriptions representing both substructure abstraction and identity refinement. This mechanism unifies two processes: (1) generic representations, and (2) scale dependent descriptions. The first case occurs when new constraints or features are added to refine an object's identity, much as when refining the definition of a wide-bodied aircraft to define a 747 as in ACRONYM [2]. The second case occurs when the same identity is described, but at several conceptual scale dependent representations. Marr and Nishihara [13] gave an example of this in their expansion of the "human" cylinder to "head, body and limbs" cylinders.

SMS uses one mechanism for both of these processes. Related models are linked using ELABORATION/SIMPLIFICATION statements. Subcomponents common to the linked objects may reference the same subcomponent definition, or may reference refined subcomponents. Additional property constraints as well as new models can distinguish refined models from their predecessors.

Associated with the geometric model are viewpoint dependent relationships among visible features. This information records visibly significant features, such as observability and surface ordering, for the principal distinct viewpoints associated with the object. While this information could be derived from the geometric model, the justification for including the information explicitly in the model is twofold: (1) on-line derivation is computationally expensive and (2) the theory of visual salience is not yet well developed.

The two key types of information represented are:

(1) an explicit classification of the visibility of prominent features in topologically different viewpoints, and

(2) new viewer-dependent features that only exist because the object is observed from a viewpoint, such as tangential occluding boundaries, obscuring surface relationships and tee junctions.

We now show part of the viewpoint dependent feature group for the whole drawing pin model. Only the visibility group associated with the viewpoint seen in figure 1 is given. The definition lists the two subcomponents visible from this viewpoint (the pin and base) and records that no features are tangential (i.e. possibly visible or not according to minor changes in viewpoint). The next group records the constraints between new viewpoint dependent features. The first two define TEE junctions, and list the boundary curves involved by their full path names, because the correct list of transformations from object to subobject is needed. The next two list boundaries that are occluding from this viewpoint, along with the background surfaces. The last item lists which model features (at this level) are partially obscured (the base). Finally, the model records the position constraints that define this particular viewpoint. The constraints say that the dot product between the vector from the viewer (i.e. (0,0,-1)) and the vector (1,0,0) transformed by the object position must lie between -0.9 and 0.

```
(VDFG drawing_pin
   (
      ...

      (VIS_GROUP (pin base) /* above side */

       TAN_GROUP (NONE)

       NEW_FEAT_CONSTRAINTS (
         (VPD_TEE
            FRONTCURVE pin->pin_body->
               pin_body_surf->body_tan_bnd1
            BACKCURVE
               base->base_circumference)
         (VPD_TEE
            FRONTCURVE pin->pin_body->
               pin_body_surf->body_tan_bnd2
            BACKCURVE
               base->base_circumference)
         (VPD_OCCLBND pin->pin_body->
               pin_body_surf->body_tan_bnd1
            BACKGROUND (base->base_top))
         (VPD_OCCLBND pin->pin_body->
               pin_body_surf->body_tan_bnd2
            BACKGROUND (base->base_top))
         (VPD_POFEAT base)
      )

       POSITION_CONSTRAINTS
        (((VIEWER DOTPR MAP((1,0,0))
               < 0))
         ((VIEWER DOTPR MAP((1,0,0))
               > -0.9)))
      )

         ...

      )
)
```

Several images of the drawing pin model are shown. Figure 2 shows the surface and space curve components of the model and figure 3 shows the volumetric model. While this is only a simple object, the different representations still give a reasonable characterization.

For the viewpoint dependent feature groups, a nominal object orientation from the supplied position constraints is deduced for each visibility group, which can then be drawn. Figure 4 shows the four significant visibility groups for the drawing pin.

The pin model demonstrates the main model features. Figures 5 and 6 show more complicated models used as part of our Alvey project. Figures 7, 8 and 9 show other models created using SMS: a PUMA robot model (surfaces and curves shown), an oilcan

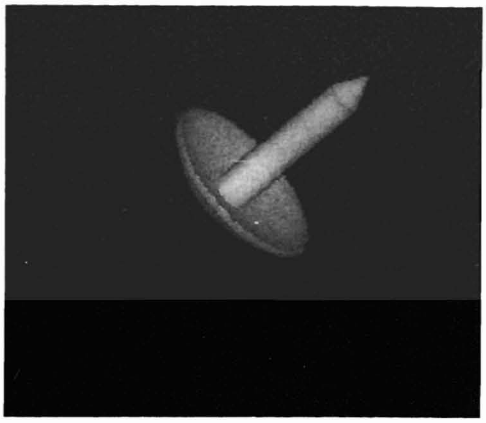

Figure 2: Surfaces and Space Curves of the Drawing Pin

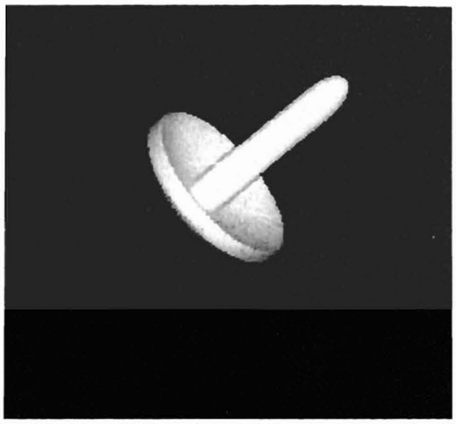

Figure 3: Volumes of the Drawing Pin

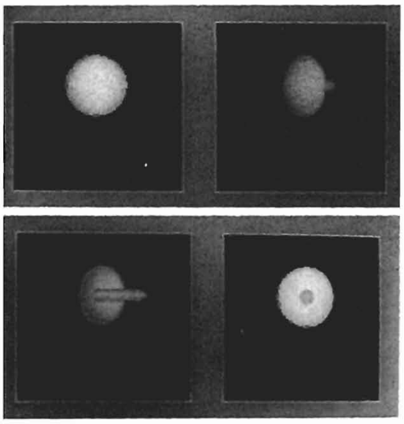

Figure 4: Visible Feature Groups for the Four Significant Viewpoints of Pin

(volumes and curves shown) and a parameterized ashtray (surfaces and curves shown).

4.1 Second Order Volumetric Primitives

Model creation using SMS's volumetric primitives revealed that these models often lacked the highly salient visual details representable using the surface and space curve primitives, although the first-order mass distribution of the features was well characterized. SMS links

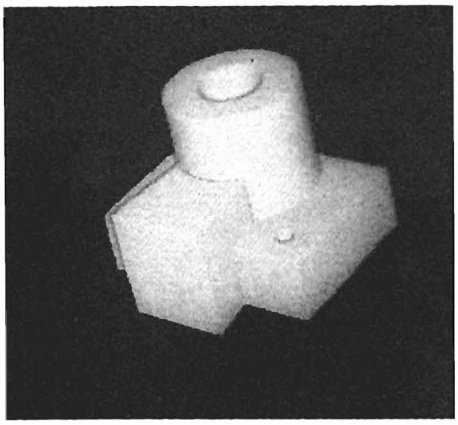

Figure 5: Surfaces and Space Curves of Widget

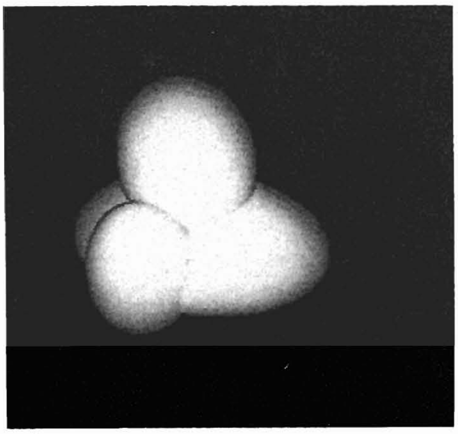

Figure 6: Volumes of the Widget

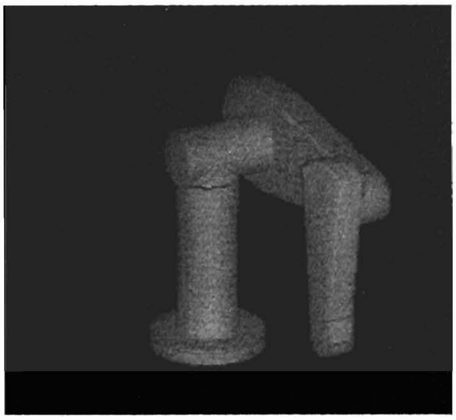

Figure 7: Surfaces and Space Curves of Robot

models by ELABORATION and SIMPLIFICATION, where linked models may have radically different structures (as in replacing a hand with 5 separate fingers by a BLOB), but exactly what the model differences are has not been clear. Hence, the motivation for the second-order primitives is to introduce new capabilities needed for having alternative conceptual scale object representations.

The major deficiencies were not having primitives for small intruding (or negative) features, like holes, and small extruding (or positive) features, such as bumps. This section introduces eight second-order volumetric primitives [6] (four small positive extruding and four negative intruding) that add detail to models, as might be required in a recognition scheme that used conceptual scale, and provides a taxonomy for them. These features increase the 'sensitivity' of the modeling scheme [12].

The extensions to SMS given here are related to the set-theoretic or constructive solid geometry [14] approach. However, here, the intent is to represent only volumetric features that can be directly and easily identified from 2 1/2D data. The primitives are also related to the shape features identified by Jared [9] and Kyprianou [11] (protrusions or depressions, which were further refined to slots, holes and pockets). Here, their role is related to part function and manufacturing method.

4.1.1 Positive Second-Order Volumetric Features

These are small extrusions modifying a major volumetric feature that do not merit a first-order feature description. They can be classified according to their having one, two or three primary directions of extension and are shown schematically in figure 10.

The first one dimensional positive feature is the **SPIKE**, which is a feature that sticks out from a volume and possibly bends (figure 10a). It is defined primarily by its length and bend curvature.

The second one dimensional positive feature is the **RIDGE**, which is a feature that lies on the surface of a volume (figure 10b). It is again defined primarily by its length and bend curvature.

The two dimensional positive feature is the **FIN**, which represents something like a **RIDGE**, but extends substantially out of the object (figure 10c). It is defined primarily by its length, height and bend curvature.

The three dimensional positive feature is the **BUMP**, representing a small hemi-ellipsoidal extrusion from a volume (figure 10d). It is defined by its three radii of curvature, given as height, major_radius and minor_radius.

4.1.2 Negative Second-Order Volumetric Features

These are small intrusions modifying a major volumetric feature. They differ from the positive features in that they cannot be approximated by SMS's current volumetric primitives. They sculpt out portions of volumetric primitives, rather than add minor extensions.

The negative second-order volumetric features can be classified according to their having one, two or three primary directions of extension and have an exact correspondence with the positive second-order features. The features are the **HOLE**, **GROOVE**, **SLOT** and **DENT**, which correspond to the SPIKE, RIDGE, FIN and BUMP, as shown in figure 11.

4.1.3 Examples

Figure 12 shows examples of an object containing SPIKE, RIDGE, FIN and BUMP features. Figure 13 shows examples of an object containing HOLE, GROOVE, SLOT and DENT features. Since the first-

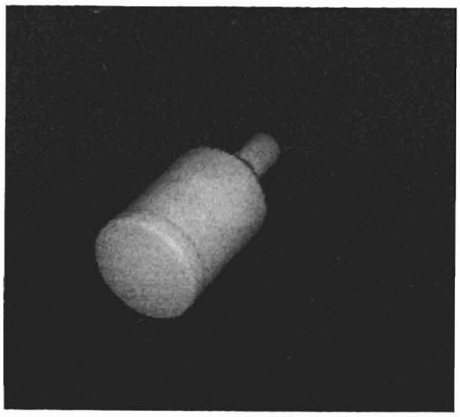

Figure 8: Volumes and Curves of Oilcan

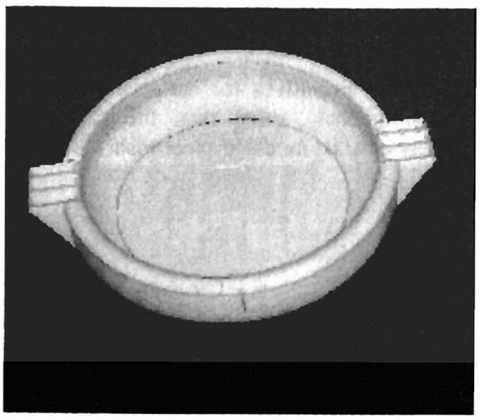

Figure 9: Surfaces and Space Curves of Ashtray

order feature in both cases is only a STICK (e.g. the largish cylindrical shape), the second-order features clearly add important distinguishing detail. Figure 14 show the volumetric model of the widget with the second order features.

Figure 12: Example Using Positive Features

5. Discussion

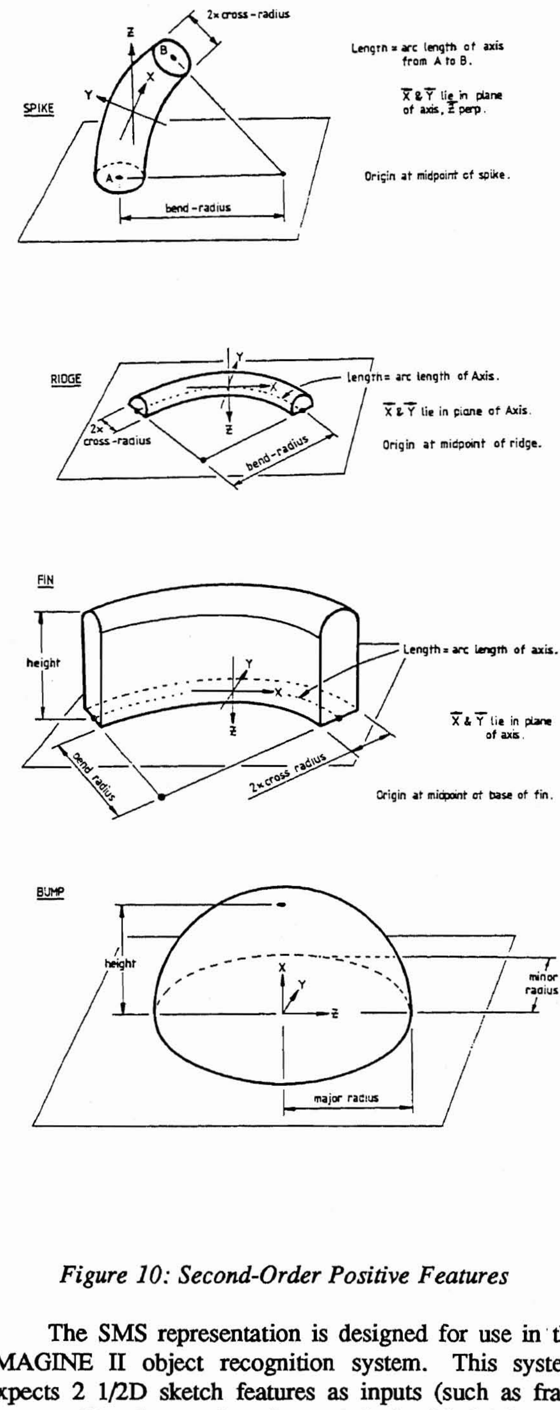

Figure 10: Second-Order Positive Features

The SMS representation is designed for use in the IMAGINE II object recognition system. This system expects 2 1/2D sketch features as inputs (such as fragmentary 3D edges and surface patches). Model invocation occurs as described in section 2, in a network created from the SMS model and the image structure. High plausibility nodes are selected for model directed matching. These nodes provide direct linking of model to data features, including several subcomponent pairings (which are then used for initial position estimation). Additionally, invocation specifies a rough object orientation, which indexes a viewpoint dependent feature group.

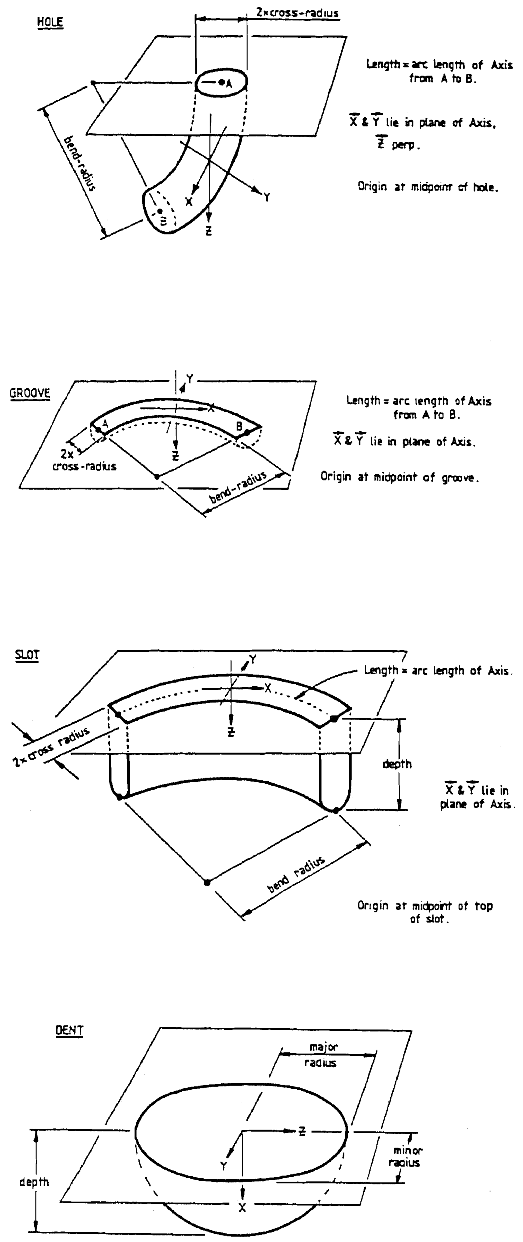

Figure 11: Second-Order Negative Features

Using this information, high performance object recognition can be quickly achieved:

- reference frames can be established from model-data feature pairings
- all visible features can be searched for, using predicted image positions and 3D constraints (from the oriented model and visibility lists),
- multiple feature fragments can be associated with oriented model features,
- back-facing and self-obscured features can be ignored,
- features obscured by unrelated objects can be verified as not visible by comparing the predicted 3D scene location with observed closer surfaces and
- viewpoint dependent features can be used as additional corroborating evidence.

The second-order features introduce the problem of whether a feature should be represented as a first or a second-order volumetric feature. For example, a nose on a face seems like a second-order BUMP relative to the whole face, but an arm on a torso is probably instead a first-order STICK extension. We hypothesize that the first-order features will be useful for broad class identifications and rough location, and the second-order features will refine subclass identifications and locations (much as in ACRONYM [2]).

There are some object representation problems that SMS does not attempt to solve:

(1) there are no primitives for surfaces whose shapes vary continuously, other than the cone - hence these can only be modeled piecewise.

(2) natural object shapes exhibit controlled irregularity, which is not represented.

(3) no metafeatures are included - such as a row of dots.

Typical objects have many (e.g. 50+) characteristic views, when viewed as a whole, potentially requiring an enormous model. To overcome this, we exploit the hierarchical structure decomposition of the objects: the viewpoints for the structured object will be classified according to the visibility of the subcomponents, rather than according to the features of the subcomponents. Further, only the significant views are represented, with minor variations remaining unmodeled.

SMS does not presume that all of an object's modeled aspects will be directly specified by the model creator. Rather, it advocates what a recognition-oriented object model should contain, irrespective of how the model is created. From the geometric model, some of the other information may be automatically derived. These include the properties of various features, such as surface curvature, angular relationships between pairs of surfaces or curve length. Some open problems are how to generate the generic and scale relationships represented by the elaboration mechanism, how to partition the features into a subcomponent hierarchy, and how to deduce unconstrained, partially constrained or variable relationships from a few observed instances of the objects.

6. Summary

SMS is an object representation system motivated by the requirements of object recognition instead of object depiction. Strongly visible features and relationships were represented as distinct symbolic primitives, which allow direct symbolic matching. It still has a structural flavor, however, and can produce reasonable pictures of objects.

Figure 13: Example Using Negative Features

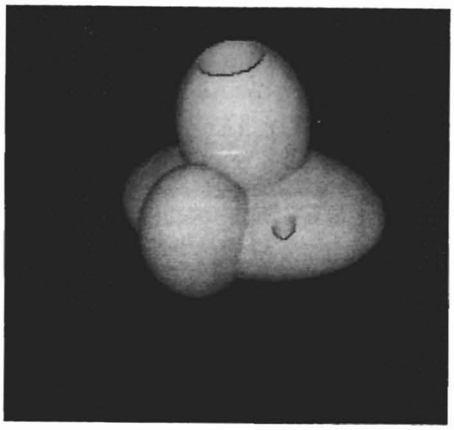

Figure 14: Second Order Widget Features

The key novelty of this representation is its integrated use of multiple alternative representations - allowing curve, surface or volumetric entities at the primitive level and refined alternative models at all levels. The advantage of these is that recognition is then achievable using a variety of evidence or recognition pathways. The alternative model mechanism combines both generic and descriptive refinement mechanisms.

It uses symbolic primitives that suggestively characterize the object and its shape, using properties that are easily extractable from image data. The result is that the object is described not literally, but instead by the character of features useful for its recognition.

The primitives are chosen for representation of solid and laminar objects with smooth surfaces (and is not restricted to the polyhedral world). This allows surface and volumetric shapes to be represented instead of simply orientation discontinuities and vertices, or infinitesimal surface patches.

Viewer-centered properties based on feature visibility and occlusion relationships are provided. They link directly with the object-centered descriptions, allowing access to viewpoint independent models from observed features.

Acknowledgements

Funding for the research described herein was under Alvey grant GR/D/1740.3. The ideas in this paper benefited greatly from discussions with A. P. Fothergill, J. Aylett, J. Frisby, M. Gray, J. Knapman, J. Mayhew and M. Orr. J. Aylett, M. Orr and F. Seytter helped considerably with the software and modeling.

References

[1] Bolles, R.C., Horaud, P., Hannah, M.J., "3DPO: A Three-Dimensional Part Orientation System", Proc. 8th Int. Joint. Conf. on Artificial Intelligence, pp 1116-1120, 1983.

[2] Brooks, R. A., "Symbolic reasoning among 3-D models and 2-D images", Artif. Intel., **17**, pp285-348, 1981.

[3] Faugeras, O.D., Hebert, M., "A 3-D Recognition and Positioning Algorithm Using Geometrical Matching Between Primitive Surfaces", Proc. 8th Int. Joint. Conf. on Artificial Intelligence, pp 996-1002, 1983.

[4] Fisher, R. B., "SMS: A Suggestive Modeling System for Object Recognition", University of Edinburgh, Dept. of A.I. Working Paper 185, 1985.

[5] Fisher, R. B., "From Surfaces to Objects: Recognizing Objects Using Surface Information and Object Models", PhD Thesis, University of Edinburgh, 1986.

[6] Fisher, R. B., "Modeling Second-Order Volumetric Features" Proc. 3rd Alvey Vision Conference, Cambridge, pp79-86, 1987.

[7] Gray, M. "Recognition Planning From Solid Models", Proc. 1986 Alvey Computer Vision and Image Interpretation Meeting, Bristol, 1986.

[8] Grimson, W.E.L., and Lozano-Perez, T., "Model-Based Recognition and Localization from Sparse Range or Tactile Data", Int. J. of Robotic Research, Vol. 3, No. 3, pp 3-35, 1984.

[9] Jared, G. E. M., "Shape Features in Geometrical Modeling", Unpublished report(?), Dept. of Engineering, Univ. of Cambridge.

[10] Koenderink, J. J., van Doorn, A. J., "The Shape of Smooth Objects and the Way Contours End", Perception, Vol 11, pp 129-137.

[11] Kyprianou, L. K., "Shape Classification in Computer-Aided Design", PhD thesis, Univ. of Cambridge Computer Lab, 1980.

[12] Marr, D., "Vision", pubs: W.H. Freeman and Co., 1982.

[13] Marr, D., Nishihara, H. K., "Representation and Recognition of the Spatial Organization of Three Dimensional Shapes", Proc. Royal Soc. London, B200, pp269-294, 1978.

[14] Requicha, A. A. G., Voelcker, H. B., "Constructive Solid Geometry", Univ. of Rochester, Production

Automation Project, memo TM-25, 1977.

[15] Shapiro, L.,, Moriarty, J., Mulgaonkar, P., Haralick, R., "Sticks, Plates, And Blobs: A Three-Dimensional Object Representation For Scene Analysis", AAAI-80, Aug 1980.