SMS: A SUGGESTIVE MODELLING SYSTEM FOR OBJECT RECOGNITION

Robert Fisher

D.A.I. RESEARCH PAPER NO. 298 *

* This is a modified version of Working Paper No. 185

Paper presented at *1986 Alvey Computer Vision and Image Interpretation Meeting.*

Also submitted to *Image and Vision Computing*

SMS: A Suggestive Modeling System for Object Recognition

Robert Fisher
Department of Artificial Intelligence
University of Edinburgh
5 Forrest Hill
Edinburgh EH1 2QL

Abstract

This paper defines an object representation system motivated by object recognition instead of object depiction, representing strongly visible features and relationships of non-polyhedral manmade objects. The representation system integrates variably sized or placed curve, surface and volumetric structural descriptions in a subcomponent hierarchy. Generic, alternative (e.g. refinement) and viewer dependent descriptions are also represented.

Keywords: object models, recognition

1. Introduction

SMS [1] is an object representation system motivated by the requirements of recognition instead of depiction. Hence, it is designed for model invocation, reference frame estimation and matching (roughly as in IMAGINE [2], only with many extensions). It represents strongly visible features and relationships of non-polyhedral manmade objects, integrating curve, surface and volumetric structural descriptions in a subcomponent hierarchy.

The central principles of the modeling philosophy are:

- 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 [3]).

- Both data and model features must be similarly segmentable so there is something to correspond and must be symbolically describable for efficient comparison. Matching requires corresponding features, though the types need not be identical (e.g. parallel tangential boundaries to cylinders).

- Marr's uniqueness criteria is to be slightly relaxed in the proposed modeling system, as there may not be a canonical description. Alternative object representations are allowed to cope with both incomplete descriptions and scale-based description change. (E.g. a pencil could be represented as either a cylinder or a collection of six elongated planes.)

- The models are suggestive rather than literal. 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. Surface splines would give a literal model of a nose; a suggestive model drawn (and labeled) would be something like:

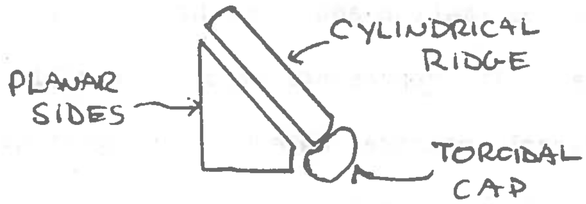


Figure 1: A suggestive model of a nose

- Marr's scope, stability and sensitivity criteria ([3]) still apply.

This paper 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. A human may define the geometrical portions of the model, with a reasoning process with visual understanding deriving the relationship and viewpoint dependent portions.

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 object model.

2. Requirements on SMS

Object representations are required for the following purposes:

- visible object features and their configurations are needed for model invocation,

- object feature and relationship descriptions are needed to constrain model-to-data matches and object 3D placement,

- predicted feature relationships are also needed to inform on what is visible from a given viewpoint.

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

The geometrical body model also introduces a uniform level of description suitable for a large class of objects (especially man-made). Rather than have the model creator decide what are the relevant features needed for recognition, the system can decide from the model itself, assuming the descriptive adequacy of the modeling system.

For matching, the geometrical model should:

- represent strong edges,

- make surface information explicit, because surfaces are the primary visible features,

- make volumetric information explicit, because volumetric relationships can be present in the data when usable surface information is not,

- be able to represent solid and laminar objects,

- have three dimensional, transformable representations for understanding appearance from arbitrary viewpoints,

- have geometrical part-whole relationships,

- allow partially constrained size and placement relationships, and

This information should be represented explicitly or be easily derivable, for matching efficiency.

Model invocation is based on accumulating plausibility through direct and indirect evidence for objects, mediated by the associations between objects [2].

The possible existence of related objects provides indirect evidence for the object, through the subcomponent, supercomponent, subtype (specialization) and supertype (generalization) relationships. The component relationships are implicit in the subcomponent hierarchy of the geometrical model, but the extra-geometric generic relationships are given separately.

Direct structure property constraints provide direct evidence for an object. Individual structure constraints specify the value ranges that are acceptable for different attributes of objects. Pairwise evidence constraints specify the 3D spatial configuration of the object features. Examples of these two types of information are the expected area of a surface and the angles at which two surfaces meet. This information is derivable (in principle) from the geometrical object model, but is made explicit in the model for efficient invocation and matching.

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. SMS's Relationship to Previous Modeling Systems

Its closest relatives are the modeling from ACRONYM [4] and IMAGINE [2]. The hierarchical reference frame and volumetric method used in SMS follows ACRONYM, though the primitive solids used are not generalized cylinders. The volumetric primitives of Shapiro et. al. [5] were chosen to represent the essential character and relationships of solids.

IMAGINE used surfaces as its primitives in a subcomponent hierarchy to make explicit the shape of individual objects. It used primitives that directly corresponded with data entities. Surface representations also made visibility deductions easier.

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 [6], where specializations have different structural models and are linked by sub-component and generic indices to associated models.

The viewer-centered representation is based on the subcomponent group of IMAGINE, the view potential of Koenderink and van Doorn [7] and the aspect graph information proposed for the YASA representation [8]. 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 [5,]. The axis relations are similar to those given by Marr ([3]) and by Fisher and Orr [9] and express the relative size and placement of axes.

4. A brief summary of SMS

The section illustrates the contents of a SMS model through use of a drawing pin model, which is currently being used for testing other components in a 3D object recognition system (IMAGINE II). Figure 2 shows the major features of the pin.

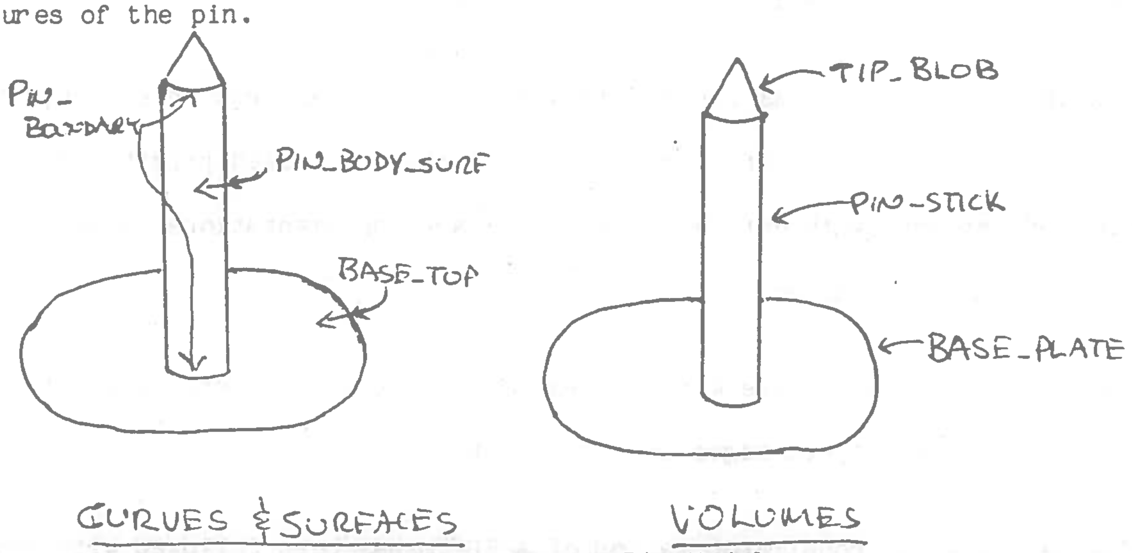


Figure 2: Drawing Pin Model

SMS's data primitives are viewpoint independent object features - having a one, two and three dimensional character - organized in an object-centered geometrical reference frame. The primitives are chosen for their visible salience. They are:

<u>space curves</u> - segmented from shape and reflectance discontinuities and 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 modelled, but this would be an easy extension.)

<u>surfaces</u> - 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 cases such as planes, cylinders and cones are also represented. Patch boundaries are nominal rather than exact and are defined using space curves.

volumes - represent extended spatial distributions, and have primarily 1, 2 or 3 directions of extension. The extensions are parameterized, so slightly distorted volumes are allowed. The three primitives are the STICK, PLATE and BLOB. The function of these primitives is to roughly characterize the mass distribution of the object without precise surface shape descriptions.

These primitives and their parameters are used because they are expected to closely correspond with descriptions taken from 2 1/2 D sketch data. The different primitive feature types are treated as alternatives, because data unpredictability may emphasize different feature types. 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 (degenerate toroidal) patch of the top surface of the pin head. The negative minor radius declares the surface to be concave. The torus 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 generic features, such as a generic robot finger used in a particular size hand.

```
(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))
)
```

surface patch - a portion of an infinite cylinder for the pin shaft's surface. Again, the pin boundaries delimit the end of the patch, and the included point indicates which section of the infinite surface is desired. Notice the variable "length" is used in the surface definition. Anywhere a value is given, an expression using variables could have been used.

```
(CYLINDER pin_body_surf
        RADIUS           0.1
        BOUNDARY_LIST   (

            ((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
            )))

        (INCLUDED_POINT (length/2,0,-0.1))
)
```

volume - the volume for the pin stick is a straight STICK. It is assumed that the CROSS_RADIUS is small compared to the length. SMS uses zero radius to denote uncurved features (i.e. infinite radius).

```
(STICK pin_stick
        LENGTH          length
        CROSS_RADIUS    0.1
        BEND_RADIUS     0.0
)
```

volume - a bent PLATE as a 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
)
```

Individual features are placed using reference-frame transformations. More complicated assemblies are formed by connecting previously defined subassemblies. The assemblies also record volumetric relationships between the solids, such as whether a stick (1D) connects to a plate (2D) in the center or the edge, and relative relationships between volume axes, such as size, orientation and placement [9].

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

- the rotation is completely constrained (but may be a variable quantity, as in a robot joint angle),
- the rotation is constrained to be a symmetric rotation about an axis, or
- 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. The point is, alternative evidence may be available for the recognition of primitive features, and this 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 DEFAULT_POSITION is for display purposes. The ASM_ALT sections separate the equivalent alternative evidence groups. One section is provided each for the curve, surface and volumetric descriptions. 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. The overall size of the subobject can be varied using the SCALE control.

```
(PRIMARY_ASSEMBLY pin_body

        (DEFAULT_POSITION AT TRANSLATION (0,0,10)
                             ROTATION RST (0,0,0))

        (VARS (NONE))

        ((ASM_ALT /* curves */

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

            (PLACED_FEATURE pin_circumference
                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 /* blobs */
                ([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 [5]). Here, the END of the pin, a STICK, is attached to the INTERIOR of the base, a PLATE.

```
(STRUCTURED_ASSEMBLY pin

        (DEFAULT_POSITION AT TRANSLATION (0,0,10)
                             ROTATION RST (0,0,0))

        (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 END_INTERIOR base))
)
```

Variables represent incompletely determined aspects of the models, such as shape, size or relative position and are bound in local contexts. The motivation for this can be shown in the example of a two-fingered robot hand. If one wants generic hand size, the variable for the size must be global to the hand. Next, joint angles in each finger must be specified independently using the joint angle variable. A problem arises when defining a single finger, and using it twice. Then, the joint angle variables of each finger instance must be separately valued, whereas the size variables of each finger instance must be the same.

The solution is to follow structured programming and define the contexts within which variables are bound. The defining context is the smallest hierarchical superobject context binding the variable. For the robot hand, the finger joint angle is defined in the context of the finger only, so has a distinct value for each finger instance. The size variable is defined in the context of the hand, but referenced in each finger subcontext, so has a distinct value in each hand instance, but the same value in each finger substance.

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 objects identity, much as Brooks did in ACRONYM [4] when refining the definition of a wide-bodied aircraft to define a 747. The second case occurs when the same identity is described, but at several conceptual scale dependent representations. Marr and Nishihara [6] gave an example of this in their expansion of the "human" cylinder to "head, body and limbs" cylinders.

SMS uses the same mechanism for both of these processes. Each object has its set of structural model definitions, and related models are then linked using ELABORATION/SIMPLIFICATION statements. Subcomponents common to the linked objects may reference the same subcomponent definition, or may reference similarly refined subcomponents. Additional property constraints as well as new models can distinguish refined models from their predecessors. An example is the simplified_pin assembly, which simplifies the pin shaft and tip to be just a single cylinder. This is defined for the case when the observer is too far to distinguish the tip from the shaft.

(ELABORATION pin SIMPLIFICATION simplified_pin)

Associated with the geometrical 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 geometrical 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, so the choice of features to represent is currently made by hand.

The two key types of information represented are:

(1) description of feature visibility according to major viewpoint - that is, a precompilation of which features are visible in topologically different viewpoints, and

(2) non-structural viewer-dependent features (ie. those that exist only

because the object is observed from a viewpoint), such as tangential occluding boundaries, obscuring surface relationships and tee junctions.

Below, we 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 2 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 viewing-caused 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 tees are formed by the two tangential boundaries of the pin against the base circumference. The next two list boundaries that are occluding from this viewpoint, along with the background surfaces. These are the two boundaries tangential to the pin with the base as the background surface. 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)))
)
        ...

)
)
```

A raycasting image generator has been developed to draw SMS models, for verifying the models. Several images of the drawing pin model are shown below. Figure 3 shows its surface-based model and figure 4 shows its volumetric model. Figure 5 shows the position of the modeled edges. While this is only a simple object, the different representations still give a reasonable characterization.

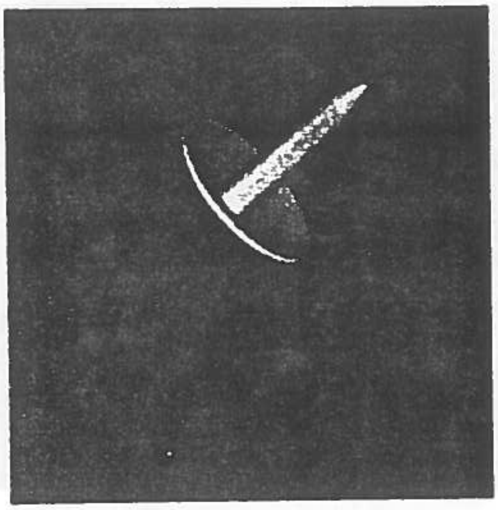
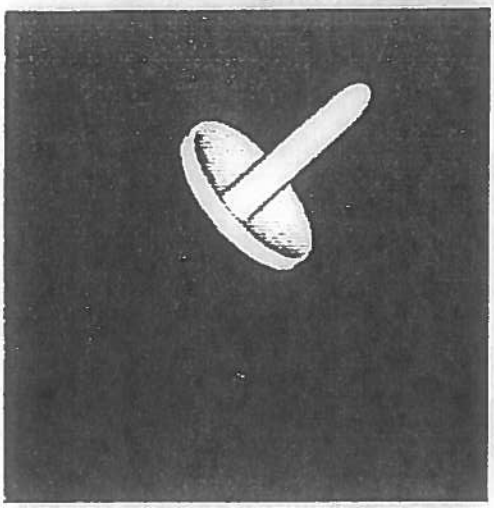


Figure 3: Surfaces of the Drawing Pin

Figure 4: Volumes of the Drawing Pin



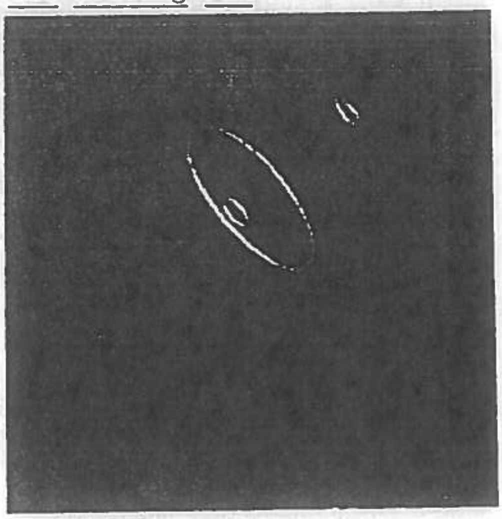


Figure 5: Space Curves of the Drawing Pin

The viewpoint dependent feature groups can also be partially verified. For each visibility group, a drawer deduces a nominal object orientation from the supplied position constraints. It then draws (1) only those features listed as being visible from the viewpoint, and (2) all features, for comparison (i.e. to show if any have been inadvertently omitted from the visible list). Figure 6 shows the four significant viewpoints for the drawing pin with the visibility groups on the top and the comparison images below.

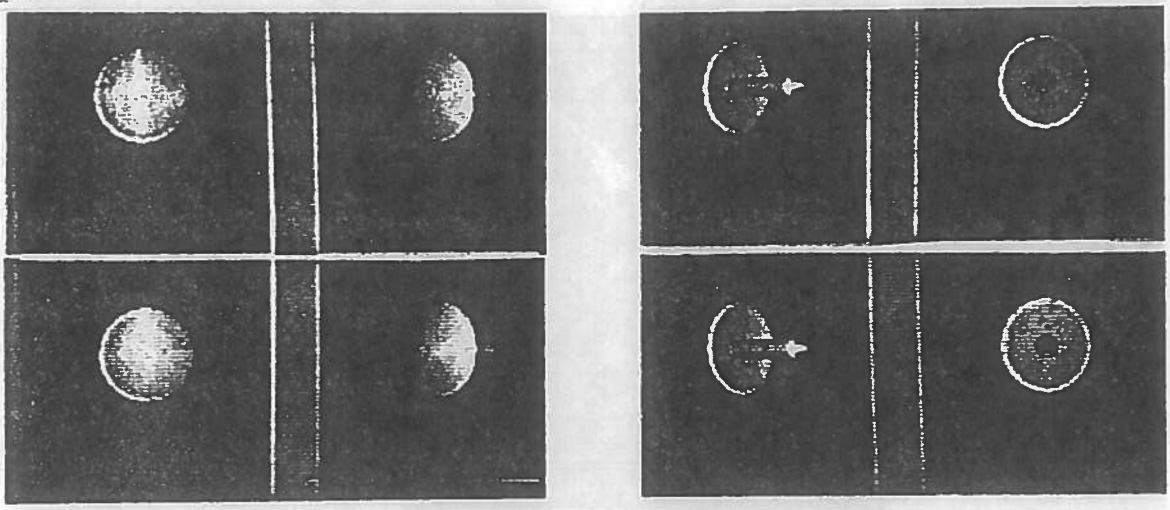


Figure 6: Visible Feature Groups for the Four Significant Viewpoints

5. Application of SMS

SMS is currently starting use in a model-based invocation and matching system (IMAGINE II). The SMS models have shown immediate usefulness:

1. The definition of the model features allow direct compilation of the invocation network [2] from the model definitions.

2. Corresponding model features are expected to be present irrespective of whether the available data feature is a curve, surface or volume.

3. The model features allows definition of partial position constraints. For example, if an image edge is paired with a tangential boundary on a surface, this provides several geometric constraints on the position the surface must hold relative to the viewer.

4. The multiplicity of model feature types allow definition of several geometric constraints on the object's position.

5. The alternative scale-based structural decomposition should allow recognition at many distances.

6. Representing surface patches by their principle curvatures allows direct correspondence with data patches.

7. The model explicitly records the primary visibility relationships when seen from the dominant viewpoints. This allows more complete image understanding (e.g. surface visibility and self-occlusion) based on equivalent views without having to recompute object visibility (e.g. through raycasting) for each model hypothesis.

6. Problems

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.

(4) the use of the alternative subcomponent decomposition method needs more evaluation in the area of non-structural (e.g. functional) decompositions.

(5) The use of the refinement mechanism for representing surface and other feature descriptions that vary as a function of scale has not been evaluated yet.

7. Conclusions

This paper presented 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 discrete matching. It still has a structural flavor, however, and can produce reasonable pictures of objects.

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 (which allow efficient discrete matching operations) that suggestively characterize the object and its shape, using properties that are easily extractable from image data. The result is that the object is not literally described (as if from a CAD model), 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 requires that surface and volumetric shapes be represented instead of simply orientation discontinuities and vertices.

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.

The variable definition method allows multiple re-use of defined subcomponents with local variables.

Acknowledgements

Funding for the research described herein was under Alvey grant GR/D/1740.3. The ideas in this proposal benefited greatly from discussions with A. P. Ambler, J. Aylett, J. Frisby, M. Gray, J. Mayhew and M. Orr. Software for producing the images was developed with the assistance of M. Orr.

References

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

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

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

[4] Brooks, R.A., "Symbolic Reasoning Among 3D Models And 2D Images", Stanford AIM-343, STAN-CS-81-861, 1981.

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

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

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

[8] Mayhew, J. E. W., "Issues Concerning The Representation of 3D Shape - A Working Paper 1", Artif. Intel. Vision Research Unit, Univ. of Sheffield, AIVRU # 1, 1985.

[9] Fisher, R. B., and Orr, M. J. L., "Spatial Configurations In The Model Invocation Process", in preparation,