MODELING SECOND-ORDER VOLUMETRIC FEATURES

Robert B. Fisher

Department of Artificial Intelligence
Edinburgh University

ABSTRACT

Model creation using SMS's volumetric primitives revealed that many models 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. This paper introduces eight second-order volumetric features that add detail to models, as might be required in a recognition scheme that used conceptual scale.

1. INTRODUCTION

SMS (Suggestive Modeling System) [Fisher 1985] is a modeling system designed for representing the salient visual features of objects, as needed by a recognition system that primarily receives three-dimensional image evidence (i.e. from a 2 1/2D sketch [Marr 1982]). SMS represents both structural and viewpoint dependent features and relationships. The primitive structural features include points, space curves, surface patches and volumes. The intent of having the three classes of primitive features is to allow alternative recognition pathways, based on alternative evidence types.

The current volumetric primitives are the STICK, the PLATE and the BLOB, which are designed for representing 1, 2 and 3 dimensions of extension. That is, a STICK represents elongated features (perhaps straight or slightly curved). It will have some thickness, but this is generally minor compared to its length. The PLATE represents flattish structures, again possibly having a slight curvature and thickness. The BLOB represents more compact structures, having similar dimensions.

Acknowledgements
This work was performed under Alvey Grant GR/D/1740.3. This paper benefited greatly from discussions with J. Aylett and M. Orr.

Larger structured assemblies are formed from these primitives using placements via reference frame transformations.

While making volumetric representations of some test objects, it appeared that these representations were impoverished compared to the surface patch and space curve representations. The volumetric features represented the first-order mass distribution of the features well, which was their purpose, but glossed over many of the obvious second-order, highly visual details. The major deficiencies were not having primitives for small intruding (or negative) features, like holes, and small extruding (or positive) features, such as bumps.

The existing SMS volumetric primitives were suitable for producing drawings containing small extrusions, for example, by using small BLOBs overlapping with the main volume. The problem is, however, representing the visual aspect of the feature for recognition purposes, rather than for drawing, and what is seen is a hemispherical extrusion, rather than half of a BLOB. Hence, another representation should be considered. A further problem is that the shapes of the extrusions do not correspond closely with those of the first-order primitives.

Jared [Jared] has recently been developing work by Kyprianou [Kyprianou 1980] on shape recognition. This work attempts to deduce the more global structure of a surface feature from a local boundary (surface, edge and vertex) description, such as the existence of a protrusion from a set of connected planes. Here, features were classified as protrusions or depressions, which were further refined to slots, holes and pockets. While the deduction method (shape grammars) is not of interest here, his proposed classifications are a subset of those in this paper. Further, the work demonstrated the importance of these second-order features (although here their role is related to part function and manufacturing method).

The extensions to SMS given here bear some relationship to the set-theoretic or constructive solid geometry [Requicha 1977] approach. However, here, the intent is to represent only volumetric features that can be directly and easily identified from 2 1/2D data. Further, the model primitives are expected to closely correspond with the data primitives. This seems to allow only disjunctive union and simple relative complement operations. Generally, CSG primitives and methods allow full boolean set operations, leading to model features that need not exist in the final object. Also, feature descriptions are not canonical, leading to recognition problems. A further problem is identifying which CSG primitive owns a given final model feature (which is needed for verifying a model).

The general CSG object modeling approach is not suitable for modeling objects for visual recognition, because the information about the visible object features are only implicit in the model. That is, one cannot easily tell what model feature a given image feature corresponds with, nor the extent of a surface feature.

This paper describes a set of small positive (extruding) and negative (intruding) volumetric features, and provides a taxonomy for them, as well as a defining syntax for use with the SMS modeler. It also considers how to represent such features (the problem being how to remove the mass associated with negative features).

In the conclusion we discuss how Marr's [Marr 1982] and Brady's [Brady 1983] criteria for evaluating representation systems apply to the new features.

2 Positive Second-Order Volumetric Features

These are small extrusions modifying a major volumetric feature that do not merit a first-order feature description. An example is a circular ridge lying around a STICK volume.

The second-order volumetric features can be classified according to their having one, two or three primary directions of extension.

The first one dimensional positive feature is called a **SPIKE**, which is a feature that sticks out from a volume and possibly bends (figure 1). It is defined primarily by its length and bend curvature. The SMS model syntax is extended for this feature to include:

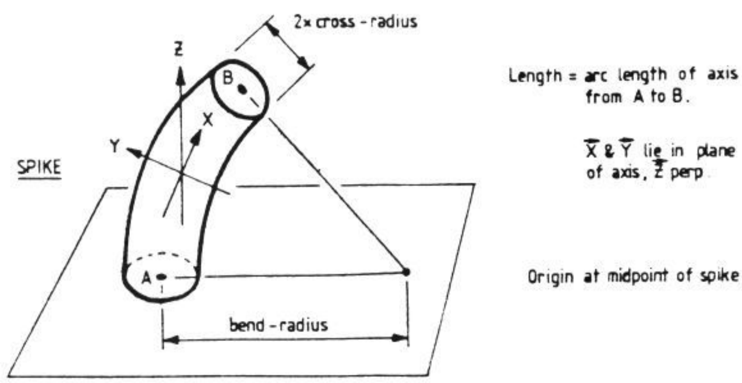

Figure 1: Definition of SPIKE Feature

```
<feature_descr> ::=
    (SPIKE <name>
        LENGTH <value>
        CROSS_RADIUS <value>
        BEND_RADIUS <value>
    )
```

The CROSS_RADIUS property defines a nominal width of the SPIKE (but is intended to be small, and is required to be less than 1/2 the LENGTH; otherwise it should be called a BUMP). If the BEND_RADIUS is 0, then the SPIKE is straight. The <name> fields accept a character string naming the feature. The <value> fields accept standard arithmetic expressions in constants, variables, binary arithmetic operators and some unary operators (e.g. cosine).

The second one dimensional positive feature is the **RIDGE**, which is a feature that lies on the surface of a volume (figure 2). It is again defined primarily by its length and bend curvature. The syntax for this feature is:

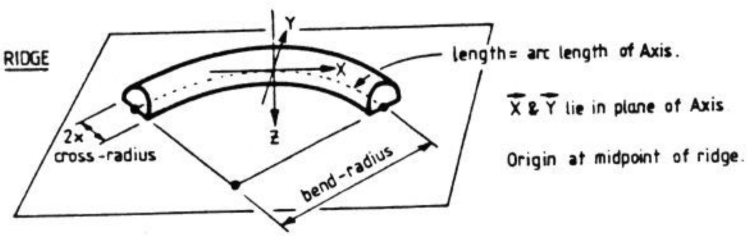

Figure 2: Definition of RIDGE Feature

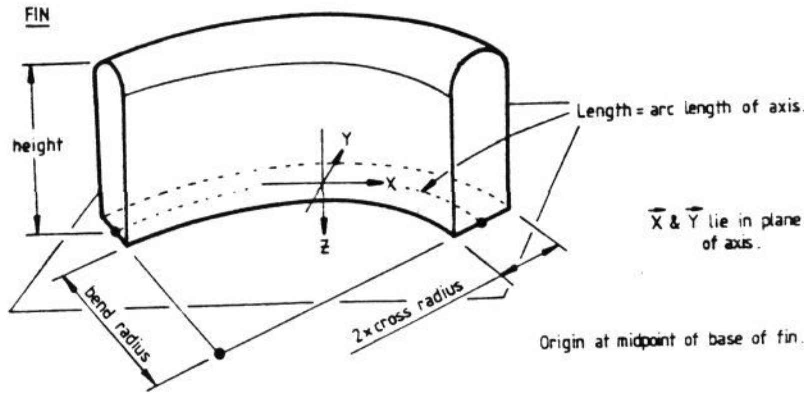

Figure 3: Definition of FIN Feature

```
<feature_descr> ::=
    (RIDGE <name>
        LENGTH <value>
        CROSS_RADIUS <value>
        BEND_RADIUS <value>
    )
```

The CROSS_RADIUS property defines a nominal width and height of the RIDGE (but is intended to be small, and is required to be less than 1/2 the LENGTH). If the BEND_RADIUS is 0, then the RIDGE is straight.

The two dimensional positive feature is the FIN, which represents something like a RIDGE, but extends substantially out of the object (figure 3). It is defined primarily by its length, height and bend curvature. The syntax is:

```
<feature_descr> ::=
    (FIN <name>
        LENGTH <value>
        CROSS_RADIUS <value>
        BEND_RADIUS <value>
        HEIGHT <value>
    )
```

The CROSS_RADIUS property defines a nominal width of the FIN (but is intended to be small, and is required to be less than 1/2 the LENGTH). If the BEND_RADIUS is 0, then the FIN is straight.

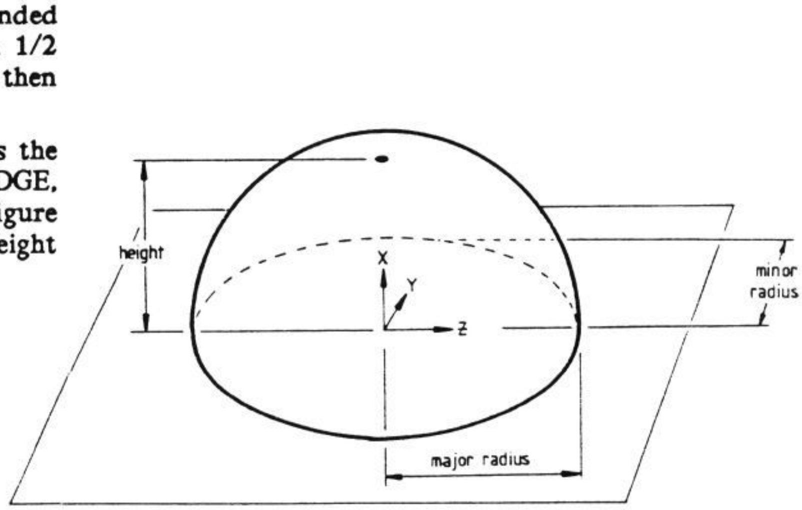

Figure 4: Definition of BUMP Feature

The three dimensional positive feature is the **BUMP**, representing a small hemi-ellipsoidal extrusion from a volume (figure 4). It is defined by its three radii of curvature, given as height, major_radius and minor_radius. The syntax is:

```
<feature_descr> ::=
    (BUMP <name>
        HEIGHT <value>
        MAJOR_RADIUS <value>
        MINOR_RADIUS <value>
    )
```

3 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. An example is a circular groove lying around a STICK volume.

The second-order volumetric features can be classified according to their having one, two or three primary directions of extension.

The first one dimensional negative feature is called a **HOLE**, which sticks into a volume and possibly bends (figure 5). This is the analogue of the SPIKE positive feature. Here, the HOLE is required to pass completely through the volume, otherwise it would be called a DENT (below). The justification for the distinction is that typical 3D image evidence is unlikely to distinguish between shallow and deep DENTs, whereas a HOLE can at least occasionally be identified by seeing completely through the object (as in the handle on a teacup). As a HOLE through a thick solid is indistinguishable from a deep DENT from many viewpoints, there is clearly some ambiguity of interpretation, but this is no different to deciding between a PLATE and a BLOB when looking along the normal to the PLATE.

The HOLE is defined primarily by its length and bend curvature. The syntax is:

```
<feature_descr> ::=
    (HOLE <name>
        LENGTH <value>
        CROSS_RADIUS <value>
        BEND_RADIUS <value>
    )
```

The CROSS_RADIUS property defines a nominal width of the HOLE (but is intended to be small, and is required to be less than 1/2 the LENGTH). If the BEND_RADIUS is 0, then the HOLE is straight.

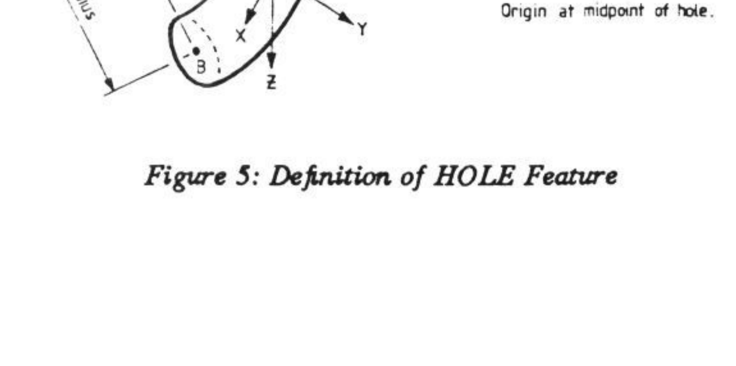

Figure 5: Definition of HOLE Feature

Figure 6: Definition of GROOVE Feature

The second one dimensional negative feature is the **GROOVE**, which is a feature that lies on the surface of a volume (figure 6). It is the negative analogue of the RIDGE and is primarily defined by its length and bend curvature. The syntax is:

```
<feature_descr> ::=
    (GROOVE <name>
        LENGTH <value>
        CROSS_RADIUS <value>
        BEND_RADIUS <value>
    )
```

The CROSS_RADIUS property defines a nominal width and height of the RIDGE (but is intended to be small, and is required to be less than 1/2 the LENGTH). If the BEND_RADIUS is 0, then the RIDGE is straight.

The two dimensional negative feature is the **SLOT**, which is intended to represent something like a GROOVE, but which extends substantially into the object (figure 7). It is the negative analogue of the FIN and is primarily defined by its length, depth and bend curvature. The syntax is:

```
<feature_descr> ::=
    (SLOT <name>
        LENGTH <value>
        CROSS_RADIUS <value>
        BEND_RADIUS <value>
        DEPTH <value>
    )
```

The CROSS_RADIUS property defines a nominal width of the SLOT (but is intended to be small, and is required to be less than 1/2 the LENGTH). If the BEND_RADIUS is 0, then the SLOT is straight.

The three dimensional negative feature is the **DENT**, which represents a small hemi-ellipsoidal intrusion into a volume (figure 8). It is the negative analogue of the BUMP and is defined by its three radii of curvature, given as depth, major_radius and minor_radius. The syntax is:

```
<feature_descr> ::=
    (DENT <name>
        DEPTH <value>
        MAJOR_RADIUS <value>
        MINOR_RADIUS <value>
    )
```

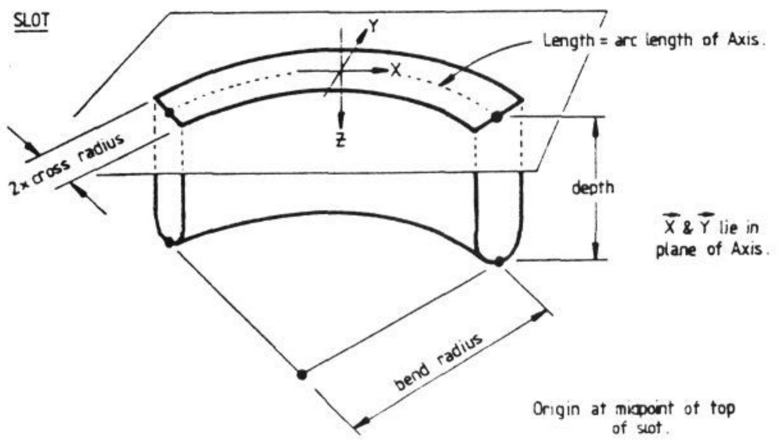

Figure 7: Definition of SLOT Feature

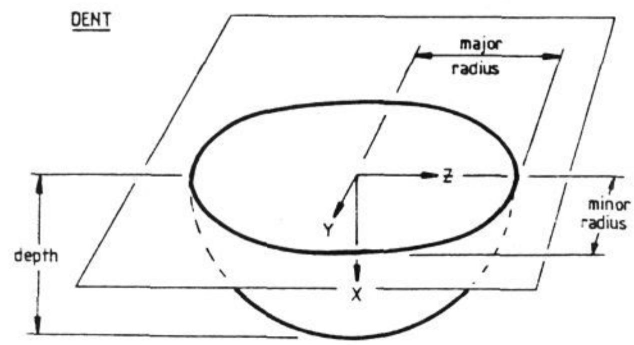

Figure 8: Definition of DENT Feature

4 Examples

Figure 9 shows examples of an object containing SPIKE, RIDGE, FIN and BUMP features. Figure 10 shows examples of an object containing HOLE, GROOVE, SLOT and DENT features. Since the first-order feature in both cases is only a STICK (e.g. the largish cylindrical shape), the second-order features clearly add important distinguishing detail. Figures 11 and 12 show volumetric models of the widget with and without the second order features.

5 Discussion

Marr [Marr 1982] proposed 5 criteria for evaluating a visual representation system: accessibility, scope, uniqueness, stability and sensitivity. While no developed processes derive volumetric representations from the 2 1/2D data yet, we believe that rough volumetric approximations are acquirable. The positive extensions are roughly of the same class, only smaller, so should also be accessible. The negative extensions are new, but seem to be simple, requiring answering only simple questions. For example:

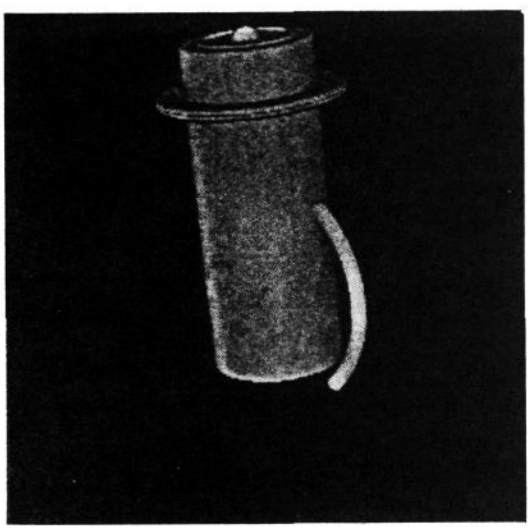

Figure 9: Example Using Positive Features

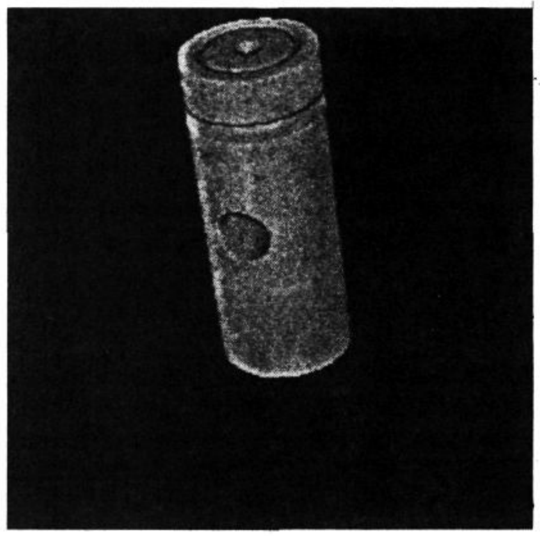

Figure 10: Example Using Negative Features

```
/* positive features */
if it is an extrusion
   then if it is shallow
        then if it is elongated
           then it is a RIDGE
           else it is a BUMP
        else if it is elongated
           then it is a FIN
           else it is a SPIKE

/* negative features */
if it is an intrusion
   then if it is shallow
        then if it is elongated
           then it is a GROOVE
           else it is a DENT
        else if it is elongated
           then it is a SLOT
           else it is a HOLE
```

Hence, the accessibility criterion should be satisfied. The modeling extensions increase the class of shapes representable, hence enhance the scope criterion. Since the above tests are mutually exclusive, and since the features are used largely for disjunctive unions and simple complementation, there is likely to be only a unique representation for any data feature, or at most only a few related alternatives. For example, if

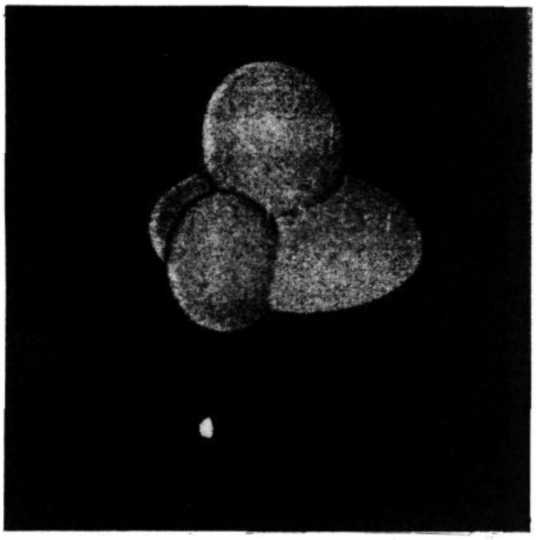

Figure 11: Widget Without Second-Order Features

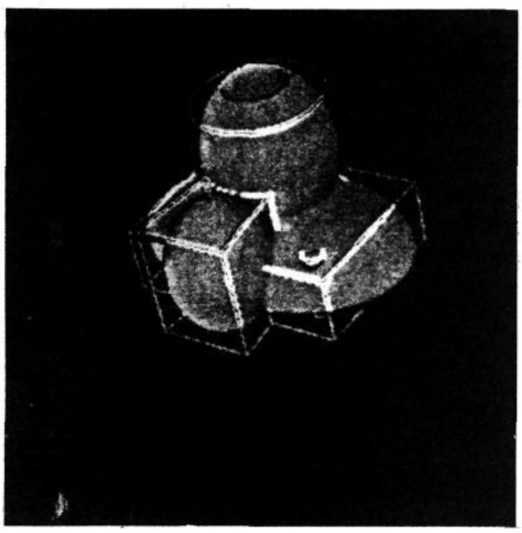

Figure 12: Widget With Second-Order Features

we see an elongated intrusion, but cannot tell its depth, we know it can only be a SLOT or a GROOVE. We argue that the first-order volumetric primitives (e.g. STICK, PLATE and BLOB) satisfy the stability criterion, while these new second-order features satisfy the sensitivity criterion, because small features are unlikely to change the class of first-order feature, yet the second-order model features can represent the new data features as necessary.

Brady [Brady 1983] added three additional criteria for representations: rich local support, smooth extension and subsumption of representations and reference frame propagation. The first and third are satisfied because: (1) the representation of the volumetric features depends on only the data that make up the features and (3) all first and second-order features have relatable reference frames. The second criterion may not be satisfied because position of the second-order feature representation is independent of the first-order representation, and hence the descriptions are independent.

The definition and use of these second-order features has raised some new questions. One concern is where should the reference frame be attached, such as on the HOLE or SPIKE. Typically, one end of this feature will be positioned relative to a known feature (e.g. the surface) and the other end is free to move according to the parameterization. Hence, one end should be at the reference frame origin. With the GROOVE feature, however, a central origin is useful when the feature lies in the middle of a surface but may vary its curvature, whereas an end origin is useful when the feature's end is fixed, but its path varies. Hence, some alternative representations having the same shape but different reference frame origins may be useful. This is only important when some of the primitive's shape parameters (such as curvature) are variable - we would like the primitive to keep its desired position without cumbersome reference frame transformation expressions. This is primarily a modeling concern, but also affects recognition, which might more easily establish a reference frame origin at the joining of the features.

A second question concerns 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. So, clearly size and relative scale plays a part here, but how is not yet clear. Perhaps a suitable criterion is: "does the description of the whole (e.g. head) depend on the two features simultaneously (e.g. nose and skull) or do the new features appear by model refinement (e.g. a smoothed head and then a head with a nose)?

Adding the positive features pose no major problems to the existing SMS modeling scheme. However, the negative features differ in that they remove material from existing positive features (i.e. the portion of a positive feature lying inside a negative feature does not exist). Following this, a BUMP lying on the inside surface of a HOLE cannot be represented. This suggests that a full constructive solid geometry (CSG) approach (e.g. [Requicha 1977]) might be adopted to overcome these problems.

The role of these new features in recognition is still hard to assess because there are not yet data analysis processes producing volumetric descriptions, let alone processes producing scale-based descriptions. 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 [Brooks 1981]). Further, the feature classes are both distinct and symbolic, promoting relational matching algorithms.

These comments converge on the real point of the extensions - the second-order primitives introduce new capabilities needed for having alternative conceptual scale object representations. SMS currently links 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). These second-order features will now allow object representations with first the basic shape and then (incrementally) finer details.

Some easy extensions could be made to the second-order primitives to allow features that twist, undulate and vary shape (perhaps like generalized cylinders). It is likely that other primitive shapes exist, too. While more effort could be spent on improving the primitives, leading to more accurate object models, we believe more effort should be spent on using the models for recognition (which we are pursuing).

The SMS models are intended to capture the "feel" of the objects, that is, their suggestive and visually salient aspects, rather than their metrically accurate character. Hence, the descriptions concentrate on "humanly" nameable volumetric features, and the second-order volumetric features defined here add new descriptions of this type to those originally used in SMS.

REFERENCES

1 Brady, M., "Criteria for Representations of Shape", in Beck, Hope and Rosenfeld (eds), *Human and Machine Vision*, Academic Press, 1983.

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

3 Fisher, R.B., "SMS: a suggestive modeling system for object recognition", Image and Vision Comp., **5**, pp98-104, 1987.

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

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

6 [Marr 1982] Marr, D., *Vision*, W. H. Freeman, San Francisco, 1982.

7 [Requicha 1977] Requicha, A. A. G., Voelcker, H. B., "Constructive Solid Geometry", Univ. of Rochester, Production Automation Project, memo TM-25, 1977.