Modelling inexact shapes with fuzzy sets

R.R. Martin

Abstract

Recently, various papers have given consideration to how *fuzzy sets* might be useful in geometric computing and solid modelling. This paper reviews this previous work and related topics, and adds further observations on the use of fuzzy sets for inexact shape modelling. It is noted that many serious problems remain to be solved. The wide range of solid modelling applications requiring the generation and recognition of inexact shapes gives a good reason for further work in this area.

Introduction

Recently, various papers have given consideration to how *fuzzy sets* might be useful in geometric computing and solid modelling. After giving a brief overview of relevant fuzzy set theory, this paper provides a review of earlier work in this area and and other related topics. This includes papers discussing the use of fuzzy sets for shape definition and image analysis, and other work aimed at describing and manipulating shapes which are in some way incompletely specified. The other part of this paper adds further observations by the current author on the use of fuzzy sets for shape modelling. These consider the differing needs for modelling inexact shapes, the stage at which fuzziness is introduced into the model, visualization of fuzzy models, recognition methods, the use of natural language, and the choice of set operators. In conclusion, it is noted that much further work is needed in this area, while if some of the outstanding problems can be solved, this approach has potentially a wide range of applications.

Fuzzy-set theory

Let us briefly consider an example to illustrate the idea of fuzzy sets. CSG, or set-theoretic modelling, relies on using regular sets to describe solid objects. Such sets can be referred to as *crisp* sets, for which the set-membership function takes on the values 1 and 0 to denote which points do and do not belong to the object. The concept of crisp sets can be extended to *fuzzy* sets, where the set-membership function can now take on any value between 1 and 0, which denotes the *grade* of membership of an object to the set. For example, consider the set of 'old persons'. The set-membership function used to determine if a given person is a member of this set might take on the value of 0 for persons less than 20 years old, rise from there linearly with age to a value of 1 for persons 60 years old, and stay at 1 for people older than that. Persons between 20 and 60 years of age can be considered to be 'old persons' to a varying degree.

A normal (crisp) set S defined over Ω has a membership function which for each member x of Ω returns 1 or 0 depending on whether or not x belongs to the set. A fuzzy set F defined over Ω has a membership function which for each member x of Ω returns a value in $[0, 1]$. This number is the *grade of membership* of x in F, and is denoted $\mu_F(x)$. A value of 0 denotes that x is not a member of the fuzzy set, a value of 1 denotes that x is a full member of the fuzzy set, and values in between denote that x is to some varying amount a member of the fuzzy set.

For example, consider the fuzzy set F of real numbers satisfying $x \gg 1$. We might define

$$\mu_F(x) = \begin{cases} 0 & \text{for } x \leq 1, \\ (x-1)/x & \text{for } x > 1. \end{cases}$$

Values of x less than 1 are definitely not members of the set; as x increases above 1 we become ever more inclined to agree that $x \gg 1$.

Various operations on fuzzy sets can be defined, some of which correspond directly to ones for crisp sets, and some of which are new. Let A and B be two fuzzy sets defined over Ω, with membership functions $\mu_A(x)$ and $\mu_B(x)$. Some of the set operators which can be defined include the following.

The **Complement** of a fuzzy set, $\bar{A}$, has membership function

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad \forall x \in \Omega.$$

The **λ-complement** of a fuzzy set, $\bar{A}^\lambda$, has membership function

$$\mu_{\bar{A}^\lambda}(x) = \frac{1 - \mu_A(x)}{1 + \lambda\mu_A(x)} \quad \forall x \in \Omega.$$

Equality of two fuzzy sets, $A = B$, occurs iff

$$\mu_A(x) = \mu_B(x) \quad \forall x \in \Omega.$$

Containment of one fuzzy set within another, $A \subset B$, occurs iff

$$\mu_A(x) \leq \mu_B(x) \quad \forall x \in \Omega.$$

The **Union** of two fuzzy sets, $A \cup B$, has membership function

$$\mu_{A\cup B}(x) = \max(\mu_A(x), \mu_B(x)) \quad \forall x \in \Omega.$$

The **Intersection** of two fuzzy sets, $A \cap B$, has membership function

$$\mu_{A\cap B}(x) = \min(\mu_A(x), \mu_B(x)) \quad \forall x \in \Omega.$$

The **Bounded sum** of two fuzzy sets, $A \oplus B$, has membership function

$$\mu_{A\oplus B}(x) = \min(1, \mu_A(x) + \mu_B(x)) \quad \forall x \in \Omega.$$

The **Bounded difference** of two fuzzy sets, $A \ominus B$, has membership function

$$\mu_{A\ominus B}(x) = \max(0, \mu_A(x) - \mu_B(x)) \quad \forall x \in \Omega.$$

The **Algebraic sum** of two fuzzy sets, $A \boxplus B$, has membership function

$$\mu_{A\boxplus B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x)\mu_B(x) \quad \forall x \in \Omega.$$

The **Algebraic product** of two fuzzy sets, $A \cdot B$, has membership function

$$\mu_{A\cdot B}(x) = \mu_A(x)\mu_B(x) \quad \forall x \in \Omega.$$

It can be shown for fuzzy sets that, while $\overline{\overline{A}} = A$, $\overline{A \cup B} = \overline{A} \cap \overline{B}$ and $\overline{A \cap B} = \overline{A} \cup \overline{B}$ are true, the relations $A \cap \overline{A} = \emptyset$ and $A \cup \overline{A} = \Omega$ are *not* true. However, the relations $\overline{A \boxplus B} = \overline{A} \cdot \overline{B}$ and $\overline{A \cdot B} = \overline{A} \boxplus \overline{B}$ are true.

For the λ-complement, $-1 < \lambda < \infty$, λ gives the degree of complementation. Thus $\lambda = 0$ gives the 'ordinary' complement, while as λ approaches -1, $\bar{A}^\lambda$ approaches the whole of Ω, and as λ approaches ∞, $\bar{A}^\lambda$ approaches the empty set. The λ-complement also satisfies the relations $\overline{A \cup B}^\lambda = \overline{A}^\lambda \cap \overline{B}^\lambda$ and $\overline{A \cap B}^\lambda = \overline{A}^\lambda \cup \overline{B}^\lambda$. It should be noted that the operators given above represent only some of the ways in which union, intersection and complement can be generalized; further parameterized classes of operators like λ-complement are discussed by Klir and Folger in[11].

Direct products of fuzzy sets can also be defined, so for example if $A \subset X$ and $B \subset Y$, the fuzzy subset $A \times B$ of $X \times Y$ is their direct product where $\mu_{A \times B}(x, y) = \min(\mu_A(x), \mu_B(y))$.

Note that the connection between union and intersection operators and max and min was observed by Ricci[20] (see below), and also by Shapiro[26] in his description of the use of R-functions for defining solids.

One particular type of fuzzy set is the *fuzzy number*, so for example the membership function of the set 'about 4.2' would be some convex function peaking at the value 4.2. If we choose to consider a particular grade of membership α or higher (the *α-cut* of the set) we end up with an interval around 4.2, and then interval arithmetic can be used to handle the resulting computations.

Review

Fuzzy-set shape modelling

The paper most relevant is that by Yamaguchi et al[33] on *probabilistic solid modelling*. They note the need to represent uncertain shapes in the early stages of design, and propose to do so with fuzzy sets, where the set-membership function takes on values other than 0 and 1 representing the *probability* that a point belongs to the given set. Probabilistic solids are created from crisp solids by using either "surface distributions" or "solid distributions". In the former case, the positions of individual bounding surfaces of the object relative to a datum surface are determined by probability density functions which fall off with increasing distance from the datum. In the latter case, the whole body is defined by such a probability density function. Such solids can be combined by either the union and intersection operators defined above for fuzzy sets, or these can be replaced by the algebraic product and algebraic sum. The advantage of the second choice is that it combines solids smoothly, blending them into each other. Other results given by Yamaguchi et al include the definition of various local modifications of probabilistic solids to produce bumps, dents and blends, and methods of visualizing probabilistic solids by drawing iso-surfaces or translucent voxels. Yamaguchi et al clearly have solid modelling in mind when describing their work. However, it is unclear whether the underlying models are based on boundary representation or CSG ideas, although the surface distribution method that they use would seem to be more applicable to a boundary model.

In a rather different approach, Pham and Yang[15] outline how to describe geometric uncertainties in manufacturing plans using fuzzy sets. As well as uncertainties in the shapes of objects caused by manufacturing tolerances, they also mention positional uncertainties caused by machine motion inaccuracies, and other sorts of uncertainty caused by missing or imprecise sensor information, and numerical and timing uncertainties, all of which can be related to geometric uncertainties. However, their approach is rather simple, and they merely discuss the use of fuzzy numbers to represent lengths and angles for describing vectors in a spherical coordinate system. They use natural language phrases to describe objects, which they note allow symbolic processing of shapes as well as numerical operations. Overall however, the examples given relate merely to two-dimensional vectors, and the paper progresses little beyond its initial observations.

Fuzzy-set image processing

Various people, starting with Rosenfeld[21], have suggested the use of fuzzy sets for image processing. A key stage in image processing is *segmenting* the image, which means grouping pixels into regions; such regions might correspond to faces of an object for example. Normally, segmentation is performed first, then other computations, e.g. finding the area of the region, are performed subsequently. Rather than segmenting the image into crisp sets at once, but using fuzzy sets instead, we can proceed with such tests while each pixel is a possible member of more than one region. To such ends, Rosenfeld discusses how topological relationships used in image processing, particularly connectedness and containment, may be defined for fuzzy subsets of image data. These ideas are further explored in[22], which also defines genus for fuzzy subsets of images. In later work, Rosenfeld and Haber[23, 25] show how to measure the diameter and perimeter of fuzzy sets. Finally, Rosenfeld[24] summarizes much of this earlier work and introduces further results on fuzzy adjacency, elongatedness and starshapedness.

Pal and Ghosh[13] add to Rosenfeld's work in several ways. Definitions for various geometrical properties of regions based on fuzzy sets are given, including area, perimeter, height, width, density and adjacency of regions; these are defined in terms of operations on membership grades for each pixel. They note that an image may be segmented by minimizing the degree of adjacency of neighbouring regions, using grey levels as membership grades; they also give alternative methods.

Chauduri[3] considers these ideas more generally and shows how to use the level sets of a fuzzy set to define, in two dimensions, shapes such as fuzzy circles, fuzzy ellipses, and fuzzy polygons. He also mentions fuzzy holes of various shapes. Although he considers translations, rotations and scalings of fuzzy objects, he only allows these to be defined by crisp amounts. In later papers[4, 5], he extends the ideas of convex and concave fuzzy sets described in his original paper. The first of these two papers gives a method for computing fuzzy convex hulls when there is a finite number of distinct membership values. Although the second paper mainly contains definitions, it does include a brief mention of the idea of a fuzzy concavity tree for building up more complex shapes.

Inexact shape modelling

Various authors have proposed methods for shape definition where the shape is generated by combining some simple elements, and the overall shape generated is controlled in a *global* way by some continuous parameter. The choice of this parameter dictates which shape is generated from a set of similar shapes. This is rather different to *parameterized modelling* where, for example, the radius of a hole may be a parameter which can be chosen by the user. Such parameters usually have *local* effects rather than global ones, or at least only affect a certain aspect of the shape produced, usually directly producing simple geometric elements with the given parameter values. Such methods differ further from *variational models* where, once more, elements of the model are parameterized, but instead of the user choosing parameter values, constraints between the parameters are expressed as equations which must solved to find appropriate values for the parameters. In the rest of this section, we will consider several global methods, and one variational method.

Ricci's early work[20] on CSG representations is interesting in that he proposed $f(x,y,z) = 1$ to be the boundary of a solid rather than the more usual $f(x,y,z) = 0$. From this definition he showed that the intersection and union of solids S_1 and S_2 can be found by computing $\max(S_1, S_2)$ and $\min(S_1, S_2)$ respectively. Finally, he noted that a smoothing approximation to these operations which blends one solid into another may be performed by replacing $\max(S_1, S_2)$ by $(S_1^p + S_2^p)^{1/p}$ and $\min(S_1, S_2)$ by $(S_1^{-p} + S_2^{-p})^{-1/p}$ where p is a positive real number controlling the amount of smoothing. Such operators are well known to fuzzy-set theoreticians as *Yager* operators[11].

Blinn[2] considered the problem of drawing molecular models. Basing his ideas on physical principles, modified for efficiency of computation, he used a *density function* $D(x,y,z) = \exp(-a_i r_i^2)$ to represent electron density at a distance r_i from an atom located at (x_i, y_i, z_i). The overall density from a collection of atoms is given by $D(x,y,z) = \sum_i b_i \exp(-a_i r_i^2)$. Finally a surface is defined by setting this density equal to some arbitrary threshold value, $D(x,y,z) - T = 0$ in order to draw the model.

Wyvill et al[31] describe a graphics system in which particle system methods are used for generating objects. Gaussian functions are used to turn the particles into "fuzzy balls" in a manner very similar to that used by Blinn. In later work[32], they discuss the generation of *soft* objects using surfaces of constant value of a scalar field which is defined using a series of key points. The main difference from Blinn's model is that each key point has a limited radius of influence.

There has been a recent resurgence of interest in the type of approach considered above, under the name of 'metaballs', and the technique has appeared in commercial software[30]. Some of the problems of bumpiness of the shapes constructed by the earlier methods appear to have been solved.

Woodbury[29] proposes a declarative approach to the construction of variational boundary-representation models. Objects like *generic* planes are used to construct models, and spatial relations such as *parallel* can be defined to operate on them. Thus, a pair of faces in such a model can be declared to be parallel even though neither of them is specified exactly. Woodbury notes particularly the importance of a solid model specification level between topology and geometry. His work is somewhat limited in that it is restricted to planar geometry; he gives no indication of how his ideas might be applied in a CSG context.

Features

Much work has been done on *features* for solid modelling. In some ways this is similar in spirit to the current work. In particular, feature recognition requires certain key aspects of the shape of an object to be identified while the exact geometry can be quite variable.

A natural language approach to feature-based modelling is discussed by Gandhi and Myklebust[7]. Features are classified into a wide range of topology groups; for each group, the shape is determined by particular defining parameters. For example, there is a group including, *inter alia*, bars, beams and billets; these are characterized by length, width and height. A hierarchical classification of shape forms is used which goes from the general to the specific, covering such aspects as general form and superficial form. Rules (inequalities and equalities) are set up for each feature type governing acceptable ranges of parameter values, and relations between the parameters. Shape-transforming adjectives and verbs are used to put additional constraints on the parameters. It would appear that when a feature is generated (as a set of B-spline patches), default values satisfying the constraints are chosen for parameters not explicitly specified by the user; relative positions and orientations are also determined by natural language phrases. Altogether, a set of 360 words for modelling features are given in Gandhi's thesis[8].

Pratt[17] discusses how form features have a wide range of potential uses, for planning manufacturing, assembly, and inspection, and for identifying important parts of objects and symmetry for FE analysis. Features represent a useful intermediate level of shape description for user input which lies between topology, which is normally not specific enough, and geometry, which can at times be too specific.

This approach requires feature generation and recognition; to perform the latter modellers must search for features amongst their internal representations. On the one hand, although there is much explicit low level data present in a boundary representation model, it is not necessarily the data required to identify features. On the other hand, the detail present in a CSG tree does not give features directly either—for example, a subtracted block might represent a slot or a hole. Some promising work on feature recognition in CSG models is being done, for example by Parry-Barwick and Bowyer[14] using Woodwark's method, but in general features recognition presents problems for current boundary representation and CSG modellers alike.

Considering feature generation, Pratt[16] identifies various problems involved with the instantiation, editing, and moving of features which are defined in terms of parameter values. These take the form of unwanted side-effects which change the topology or other global nature of the shape. Many of the issues identified by him are also relevant in the context of the range of possible shapes that a given fuzzy set might represent, some of which may not be acceptable or anticipated. This means that fuzzy-set representations must be chosen and generated with care if they are to mean what their creator intends. We will return to this topic later.

Symbolic shape processing

We start this section by noting that both Pham and Yang's and Gandhi and Myklebust's work already cited above rely on the idea of symbolic shape processing based on natural language terms.

Batchelor[1] uses PROLOG to recognize and reason about shapes of uncertain geometry in images. For example, text characters may be recognized using predicates like `about_same_height`, `above`, `apex`, `connected`, `line_end`, and `tee`. Each shape predicate is defined at a low level using basic image processing functions. For example `circular` is defined to describe a shape whose ratio of area to perimeter squared is greater than a certain value. This value is fixed for a given application, and notions like *how* circular an object is are not used, although they obviously could be defined in this manner. Batchelor notes that in general it is difficult to avoid such descriptions fitting many other shapes than those desired, again echoing the problems noted by Pratt. Batchelor gives a short list of useful English shape description terms, although some of them such as 'crossing' are more appropriate to lines in images than three-dimensional solids.

Similarly, the Thalmanns report[27] methods based on PROLOG rules to describe shapes using natural-language concepts, an example of which might be

```
larger(higher_half(myshape), lower_half(myshape))
```

but no description is given of how shapes are represented internally, nor of how the predicates work. Their paper simply gives a few examples, without giving a complete description of what predicates are available, and why these particular ones were chosen.

Ullman[28] discusses the relationships that can exist between objects which are of importance during design; many of these cover aspects other than geometric information. However, he does identify the relation of relative position as being of particular importance, noting that it may be specified in a variety of ways, from abstract terms like 'near' to precisely specified rigid body motions between reference frames. He notes that other relations which are important for determining geometry include connectivity, fits, clearances and relative paths.

Other work

Gupta and Ray[9] give a definition for *fuzzy plane projective geometry*. At present this work would seem to be of little use to fuzzy-set-theoretic modelling, but may be worth considering further.

Outstanding issues

Need for modelling uncertain shapes

Uncertainty about geometric shape of an object arises in a variety of ways. At the very highest level, we have the typical *synthesis* versus *analysis* distinction. In their work, Yamaguchi et al only considered the former.

Some applications, such as design, require the generation of new shapes. A designer may wish to *describe* a shape in top-down fashion, giving the main aspects to start with, and whilst doing this, may not want to specify the details. He may just wish to specify that the object is long and thin, for example. A method of representing incompletely specified shapes is thus required, or indeed classes of shapes which agree with such a description to a greater or lesser extent. Even in the finished design, a designer may wish to represent a class of shapes rather than just a single shape, and so there may be some uncertainty in the final result. Using a boundary representation, the *topology* of a model offers some level of shape description over and above the *geometry*; however, the topological level is usually too high to be of practical use to a designer. In conventional CSG modelling, even this higher level of description is generally lacking, and so it is fair to say that current modelling systems are very poor for describing objects whose shape is incompletely specified.

In other applications such as computer vision and feature-based process-planning, we need to *recognize* existing shapes. Here, we have some idea of what an expected or sought-for shape will look like, but not the exact form that a given instance will take. What is needed is a shape description for the whole class of shapes to be recognized, rather than for any individual shape. The variations between the individuals, and the fact that these cannot necessarily be predicted in advance, lead to the need for inexact shape models for recognition purposes. While Yamaguchi et al use the membership function to denote the probability that a given point lies inside a synthesized shape, most books on fuzzy-set theory emphasize that in general fuzzy membership functions *do not* represent probabilities. In some ways, this ties in better with the approach taken in analysis, where we may be more interested in to what extent a shape is long and thin, rather than the probability of it being long and thin.

Let us briefly consider further typical areas in which the geometric modelling of uncertain shapes may be useful. Perhaps one of the most obvious is description of shapes with attached *tolerances*. Here the uncertainty represents acceptable errors in manufacturing, and we typically wish to compare a precisely measured input shape under test with a stored model which describes allowable variations. In a similar vein, but with rather less precise ideas of what the input object should look like, are various *object recognition* requirements, where we may wish to find 'aircraft shapes' in aerial views of airports, for example. Computer-aided design systems which employ *feature recognition* have similar needs, although in this case parts of objects are of interest, rather than whole objects.

Design with current CAD systems requires the user to work with exact shapes towards achieving a finished object. Such an approach is not necessarily ideal. For example, a designer may know that he needs a long rod with functional parts on each end of it, but having initially to choose a rod with a circular, square or other cross section may be an irrelevance; also, the length of the rod may not be known until the functional parts have been defined. Allowing *inexact designs* where the rod is just described as a 'long rod' may be useful. A completely different set of applications may also find inexact shapes useful as representations for computing various properties of exactly defined shapes. One such possibility is that *Voronoi-like diagrams* which indicate which object is nearest to a test point (using some approximate distance metric) could be computed by taking the exact shapes, and "growing" them at the same rate by an uncertain amount until one of them reaches the test point.

In some cases in design it is advantageous to leave certain aspects of the design as variable, while other aspects are fixed, resulting in *parameterized models*. Such models do not normally reflect any uncertainty, as there is the implicit expectation that before the model is used for anything, values will be supplied for these parameters. Note that parameterized models can allow for classes of objects of widely varying generality—consider the differences between spheres of varying radii, ellipsoids with varying axes, and superellipsoids of varying powers. More or less arbitrary shapes can be composed from models if these have enough parameters.

Similar remarks about uncertainty apply to *variational models*, the difference being in this case that the values are determined by solving constraint equations. Hoffmann[10] clearly points out the dangers of the user interacting with specific examples of variational designs and then changing the parameters. Software which chooses the wrong solutions of large sets of constraint equations can result in designs which are geometrically very different from those originally intended; this can be difficult to avoid even when heuristics are used to guide the choice of solution. The uncertainty arising in such cases is more to do with topology and configuration than with shape. The issues Hoffmann raises are just one example of the way in which hidden constraints and assumptions exist which we take for granted when dealing with imprecise concepts, but which may be very difficult to build into software in a way which produces the expected results all or even most of the time.

Finally, it is worth noting that uncertainty can take on several forms. Thus, for example, the radius of a hole may have a range of acceptable values $[10, 20]$, or a discrete set of acceptable values $\{5, 10, 15, 25\}$. While the latter can in principle be dealt with formally by fuzzy-set theory using δ-functions, we will not consider this type of uncertainty further here.

How to include the fuzziness

Let us consider CSG models built with fuzzy sets. The fuzziness can be introduced at various places. We need to consider carefully at what stage to introduce the uncertainty to produce any overall desired inexact shape. Individual halfspaces can be made fuzzy by replacing zero in $f(x, y, z) = 0$ by a fuzzy number. Other parameters in $f(x, y, z)$ may also be replaced by fuzzy numbers, although the algebraic parameters normally have little direct geometric significance. The transformations used to position, scale and orient nodes of the CSG tree before they are combined may contain fuzzy numbers. Any crisp CSG set, from a primitive upwards, may have fuzziness added to it afterwards, for example by basing the membership grade function on offsets. Of these possibilities, Yamaguchi et al only identified the last. On the other hand, Pham's work essentially only considered transformations, rather than shapes.

Let us suppose we want to represent a 'short bar', which we can consider as a one-dimensional problem. Let us fix the left-hand end. We can then use the idea proposed by Yamaguchi et al directly, where we have a membership grade which drops off with distance from the fixed end—as we get further from it, the points are less likely to be in the bar. If we now ask the deceptively similar question about how to represent a 'long bar', we run into trouble. The membership grade cannot fall off as we go away from the fixed end, as this would mean that a shorter bar is more likely than a long bar. On the other hand, the membership grade cannot be less for points nearer the fixed end—the very existence of a point in the object at the variable end guarantees that points in the fixed end of the bar must be present. Thus the approach of having a membership function defined pointwise over all space does not work in this case. (The given argument does not rely on fixing one end of the bar, and its centre could be fixed instead, for example.) Long and short bars can both be described, successfully, however, by (a) having a bar of standard length, to which is applied a fuzzy scaling transformation to adjust its length, or (b) applying a fuzzy constraint to the positions of the ends of the bar.

Even for simple objects, the nature of uncertainty about a shape may take on several forms. Thus, we may have a well-defined form of imprecise *size*, such as a 'large sphere'. We may also have a shape whose size *and* form are ill-defined, such as an 'almost spherical object'. Such fuzziness may need to be expressed as constraints relating various properties of the object: such as the surface area and volume, in this example. If such concepts are to be handled, then, even in the simplest cases, variational systems will be required which can handle not only equations, as at present, but also inequalities. More complex relations of the 'almost spherical' type will not only require these capabilities, but also sophisticated optimization methods which can search for objects which satisfy the constraints to varying degrees.

One such approach to describing different types of variation of objects is to represent the variations as separate size, form and position tolerances, as described in Requicha's work[18] on geometric modelling of tolerances. These have the disadvantage that they are not independent. The simplest to deal with are position tolerances; assuming a test object has been correctly positioned and oriented, all points of each surface of the test object must lie within a given distance of their correct positions. Clearly, this can easily be modelled by the methods given above for offsetting, or a bar with a nominally circular cross-section can be modelled as a bar of fuzzy radius, for example.

In the case of size tolerances, all points of the test object must lie within a set distance of the correct reference surface, but that surface can be arbitrarily positioned and oriented in space. However, the simple solution here, of applying a fuzzy transformation matrix to the fuzzy bar defined in the first step, does not work if done in the obvious way. The problem is that all points under comparison must be subject to *the same* transformation, which they will not be if they are tested one by one. An even simpler form of this issue arises when considering the question "How does one represent an *exact* circle of unknown radius using fuzzy-set methods?" As we have already noted, using $x^2 + y^2 - r^2 = 0$ where r is a fuzzy number represents a set of points which almost lie on a circle; this is not the same as a set of points which lie on an exact circle of unknown size. One possibility is to include a constraint, as mentioned earlier, enforcing the appropriate relation between the area and perimeter of the shape. Such constraints will need to take the form of *invariants*, i.e. quantities which remain fixed for classes of objects. Invariants are already widely used for recognition purposes in computer vision[12]. Nevertheless, it seems likely that formulating suitable constraints in general may be very difficult.

Considering the final case of form tolerances, all points must lie within a set distance of some arbitrary surface of the correct form—a sphere, for example, even if it is of quite the wrong size. This surface can again be arbitrarily positioned and oriented in space. The approaches mentioned for representing size tolerances are applicable here; suitable constraints will be even more complex.

Finally, we note that Requicha also discussed the ideas of maximum and least material conditions for an object to be within tolerance, whereby the original object is offset inwards and outwards on a feature (face) by feature basis to obtain bounds on acceptable shapes. This is clearly very similar to (and predates) the surface distribution method of Yamaguchi et al. Requicha initially used boundary representations to describe tolerances in this way, although a later paper showed that the use of CSG representations was also possible[19].

Choice of set operators

Both Yamaguchi et al, by using the algebraic product and algebraic sum, and Ricci, in using Yager operators, discovered that smooth blending could be obtained by replacing union and intersection with other more general fuzzy-set operators. However, no one appears to have considered the use of the bounded sum and difference operators, the λ-complement, and other parameterized fuzzy-set operators. Work is needed to investigate a wider range of operators to see what their effects are, and when these various different operators might be most appropriate. Indeed, some of these operators may well prove useful for generating blends in crisp-set-theoretic modelling too.

Using natural language

One way of describing inexact shapes is through the use of natural language; this method would seem to be well suited to non-technical users. Most work to date using natural language for geometric computing has been based on symbolic reasoning. Hybrid systems which use symbolic reasoning for the more abstract levels of reasoning about shape and fuzzy-set-based models for more detailed work are likely to have an advantage over any system based on just one of these components. When using natural language shape descriptions interactively, it will certainly be a good idea to keep the original input form, as the method of user working is likely to depend on creating many possibilities until the right one is generated: make a long box with a slot, make the slot broader, make the slot very long.

Generally, a wide range of natural language terms may be useful in constructing a shape description. Note that form and, say, size are not necessarily defined by nouns and adjectives respectively, as the use of 'box' and 'long' above would suggest: for example, the adjectives 'hollow' and 'blunt' describe form. Often the same basic word can be used in several different grammatical forms, for example, 'circle', 'circular', 'circularity'. Thus, rather than classifying words as parts of speech, it is more useful to classify terms based on their effect on shape. This section gives lists of possible terms for natural-language-based shape description (see Tables 1 and 2), and

Modifiers

- almost
- extremely
- fairly
- moderately
- negligibly
- partially
- partly
- roughly
- significantly
- slightly
- very

Simple Transforms

Overall Scale

- big
- bulky
- chunky
- compact
- enormous
- giant
- huge
- large
- little
- massive
- minute
- small
- tiny
- vast

Scale in One Direction

- broad
- compressed
- deep
- elongated
- extended
- fat
- high
- long
- low
- narrow
- plump
- short
- slender
- slim
- squat
- stocky
- stout
- tall
- thin
- tubby
- wide

Orientation

- horizontal
- level
- reclining
- sloping
- tilted
- upright
- vertical

Position

- centred
- concentric
- radial

Shear

- slanted

Relative Transforms

Scale

- bigger
- longer

Orientation

- aligned
- along
- diagonal
- inclined
- inverted
- longitudinal
- mirrored
- oblique
- orthogonal
- parallel
- perpendicular
- reflected
- rotated

Position

- above
- alongside
- behind
- below
- beneath
- beside
- clearance
- close
- collinear
- connected
- contained
- coplanar
- distant
- encloses
- far
- higher
- includes
- inside
- left
- lower
- middle
- near
- offset
- on
- over
- overlapping
- right
- separate
- touching
- under
- within

Parts of Objects

- axis
- back
- base
- bottom
- centre
- exterior
- front
- interior
- leftmost
- outside
- rear
- rightmost
- side
- tip
- top

Table 1: Modifiers, transforms and parts of objects

attempts to show how at least some of these might be interpreted in terms of fuzzy-set membership functions.

Firstly, however, let us consider various categories of terms that are *not* given in Tables 1 and 2. Exact shapes, such as 'sphere' are omitted, as they can obviously be adequately described by well-known methods. For the same reasons, so are exact transformations described by verbs such as 'rotate', 'position', and 'move', and set-operation verbs of precise meaning, such as 'add', 'assemble', and 'remove'. Other exact properties of objects are also omitted, including, for example, 'area', 'diameter', 'length' and 'volume'. Analogical descriptions, such as 'bullet-shaped' and 'needle-shaped', while undoubtedly useful, comprise an indefinitely long list; such shapes are likely to be specific to a given application. Functional descriptions of objects, such as 'clip', 'bush' and 'washer' normally encompass too wide a range of shape to be generally useful; a program for recognizing clips would need to know more about the function of clips than about possible geometries. Set operators which depend on understanding functionality are also omitted, such as 'fasten'.

Secondly, synonyms are included in the table, even though for example 'little' and 'small' can be considered equivalent. They are left in Tables 1 and 2 as it is intended that these tables should be fairly complete, and can act at least in part as reference tables of terms. Going a step further, words which are simple modifications of others can often be treated in a simple way. Provided we can deal with modifiers like 'very', we can trivially reduce 'huge' to 'very big' and 'enormous' to 'very very big'. On the other hand, for fuzzy sets, 'not' is not a trivial concept, and opposites need more careful thought. Table 1 starts by listing various modifiers.

When generating models, two sorts of modifiers are useful. The first takes an inexact shape and turns it into a new inexact shape. For example, 'very' can be used to turn a 'short rod' into a 'very short rod'. The second, for example, 'almost' takes an exact shape and turns it into an inexact shape, such as a 'rod almost 6 units long'. In the second case, we must replace certain information by its fuzzy equivalent, so in the case given the length of the rod would become a fuzzy number. In general, it may be much less obvious how to introduce the required fuzziness, as we noted earlier. However, standard methods of fuzzy set theory can be used at least in simple cases to deal with the first class of modifiers. A proposition

$$x \text{ is } A$$

can be modified to

$$x \text{ is } mA,$$

and to find the fuzzy set mA from A we can apply transformations such as

$$\begin{aligned}
\mu_{\text{very } A} &= \mu_A^2, \\
\mu_{\text{more or less } A} &= \mu_A^{1/2}, \\
\mu_{\text{not} A} &= 1 - \mu_A.
\end{aligned}$$

Consider, for example, a short bar as defined by a membership grade falling off with distance from a fixed end. Because $0 \leq \mu_A \leq 1$, on squaring the membership function to allow for the 'very' modification, a given point away from the fixed end has a lower membership grade for 'very short' than 'short', or alternatively, we must choose a point nearer the fixed end before it achieves the same membership grade. This accords with usual thinking if we agree that all 'very short' things must necessarily be 'short'; it ties in less well if we feel that certain things are better described as 'very short' than 'short', and presumably for this reason Pham decided not to use these standard modification methods.

Let us continue to consider the distinction between precise and imprecise meanings of natural language phrases. Many of the words other than the modifiers in Table 1 can be used in various grammatical forms, as already noted, and the meaning of each form can be quite different. Even a given form may have more than one meaning. For example, while 'long' is imprecise, 'longer' may be precise or imprecise. Given two bars of definite length, the statement that one is longer than the other has a precise truth value. On the other hand, given just a single bar, we may wish to consider the fuzzy set of 'longer bars'.

Many shape description terms can be implemented as fuzzy transformations acting within the CSG tree. Included in Table 1 are words which can be described by fuzzy scaling (along one or three axes), rotation, translation and even shearing transformations. Some words describe simple, single object transformations, such as 'long', while others describe relative transformations between multiple objects, such as 'parallel'. Many further relative transformations can be generated by taking simple ones and turning them into a comparative form, e.g. 'long' into 'longer', as noted above. Some words may fit into more than one category, such as 'aligned' which may constrain both the relative position and orientation of multiple objects, and 'contained' which may necessitate relative scaling and positioning. Such words are only listed once in the table in their most appropriate place for reasons of space, although some words such as 'connected' do not really fit well into this scheme of classification, implying more about topological than metrical properties.

Another class of words for identifying *parts* of objects, such as 'centre' and 'base', is important when specifying transformations, especially when building composite objects from primitives.

Table 2 gives many other terms which imply some constraint on local or global shape. Very loosely these can be divided into nouns which describe objects or parts of objects (features) in an overall way, and adjectives, which geometrically modify or more closely specify those shapes. We thus might refer to a 'pointed arch' or a 'circular tube', for example. These geometric modifiers will often have different interpretations for each base shape, and a large look-up table may well be necessary to take care of the possibilities, although an object-oriented programming system with inheritance might also be helpful. Again, it should be noted that various terms might well be put into several categories, so for example a 'loop' could describe a whole object or just a local feature, and 'twist' could describe a global shape, while 'twisted' might be a local or global modifier. A large amount of work is required to capture the essence of some of these shapes as definitions in terms of fuzzy sets; the work of Gandhi is of relevance here.

Visualization

Visualization of objects defined using fuzzy sets can be done in various ways. The first method, proposed by Yamaguchi et al, works for objects defined by a set-membership function associated with each point in space giving its probability of belonging to the object. Objects are drawn using transparent voxels, where the opacity of the voxel corresponds to the value of the set membership function. This

Shapes

Global

arch
axle
ball
band
bar
barrel
bead
beam
bell
belt
billet
blob
box
chip
coil
crescent
flake
gear
globule
lamina
loop
lump
oval
pad
pin
piston
plank
plate
prism
ring
rod
screw
shaft
sheet
slab
sliver
spindle
spiral
star
sweep
tube
wedge
wheel

Local

angle
aperture
apex
bend
bevel
blend
bore
boss
bowl
break
bridge
bulb
bulge
cavity
chamfer
channel
column
corner
closure
crater
crease
curl
cusp
cut
dent
depression
dimple
dip
dome
drum
elbow
fillet
fold
fork
furrow
gap
groove
hole
hook
housing
indentation
jagged
keyway
kink
lug
neck
notch
opening
orifice
peak
perforation
pocket
protrusion
ream
ripple
round
slot
spike
step
strut
swelling
taper
tunnel
twist
wall
wave
whorl
zigzag

Shape Modifiers

Global

closed
concave
convex
hollow
oblong
open
oval
rectangular
regular
round
spiral
symmetric
turned
uniform
warped

Local

acute
angular
annular
bent
blunt
bumpy
chamfered
curved
delicate
even
fair
flat
hooked
necked
obtuse
pinched
pointed
rough
rounded
sharp
smooth
stepped
straight
tapered
truncated
twisted

Table 2: Shapes and shape modifiers

is a natural extension of methods used for drawing objects based on crisp sets, although it is more tricky to associate colour information with particular faces of a fuzzy object using this approach. It also relies for visibility on this probability falling off (rather than increasing) as we go away from the centre of the object. Noting our earlier comments about using such membership functions for short bars but not long bars, this type of method will only be suited to a limited range of shapes.

A second approach to visualization, also taken by Yamaguchi et al, but also by various others[2, 20, 31], is to draw level sets of the membership function. In the simplest form, some particular grade of membership is chosen, say 0.5, and the surface with this value of μ_A is drawn. Obviously, a series of μ_A values can be drawn separately to give an idea of the range of shapes which are encompassed by the definition. Alternatively, these could be drawn as a series of transparent shells inside each other, incorporating the previous method to some extent.

Much work has considered the use of computer graphics for visualization of scientific data[6], including scalar fields. Such methods could obviously also be applied for visualizing the membership grade function throughout space.

With the more complex types of fuzzy object proposed in this paper, visualization becomes rather more tricky. Consider, for example, a plate with a circular hole of fixed size in it, where the centre of the hole is at a fuzzy position, and another plate with a hole where the position is fixed, but the size of the circular hole is fuzzy. We have already noted the problems of defining fuzzy sets for these differing cases. Even if the problems of definition can be solved, problems of visualization remain. In each case, the rim of the hole occupies an uncertain position but just drawing inner and outer limiting circles showing where the rim of the hole can lie fails to distinguish these cases. Probably the only clear way of showing objects like this, or more complex ones, is to draw a range of objects which fit the description to differing extents, together with an indication of how well each fits the description. In a design context, for example, the user can then choose one or a subset of these to make the design more concrete. However, there is the problem that if many nodes of the CSG tree contain fuzzy information, there will be a combinatorial explosion of possibilities.

Returning to simple objects, while those defined using fuzzy parameters may be relatively simple to visualize, those defined in terms of constraints may be much harder, just in terms of *generating* any objects which satisfy the constraints to varying degrees. It may be much easier to test a given object for meeting some constraints than finding any object which does. Thus it may be very tricky to see just what classes of objects recognition programs *do* recognize, a problem noticed by Batchelor.

Recognition

The input to most recognition programs is a well-defined shape, measured or otherwise observed, while the output is some description of how well that shape fits a given class (or each of several classes) of general shapes. However, in Yamaguchi's approach membership grades apply to points rather than whole shapes. Some method is needed of accumulating evidence by comparing points from the fuzzy object class with the particular input object. This evidence needs to take into account both presence of material where material is expected, and absence of material where none is expected, computed using integrals measuring volume of overlap. In order to do this successfully, we must be able to guarantee that the input object is positioned and oriented so as to give the optimum answer. It is not clear that this can be done by any methods other than trial and error. These problems will be seriously compounded when dealing with more general CSG models built up from fuzzy primitives using fuzzy transformations and fuzzy constraints; the tree structure leads to an exponential explosion of cases to consider. More work is needed to tie fuzzy-set shape modelling ideas in with the work by Rosenfeld and others on fuzzy-set image analysis.

Conclusions

Various different approaches can be and have been taken to the use of shape modelling using fuzzy sets, and these approaches are not always compatible with each other. Many serious outstanding issues exist which must be tackled before fuzzy sets can be more widely applied to inexact shape modelling. Nevertheless, a wide range of solid modelling applications require the generation and recognition of inexact shapes, and so there is good reason for this work to be done. Several key issues to be tackled have been identified in this paper.

References

[1] B.G. Batchelor,*Intelligent Image Processing in Prolog*, Springer Verlag, 1991.

[2] J.F. Blinn, A generalization of algebraic surface drawing, *ACM Transactions on Graphics* **1**,3 (235–256), 1982.

[3] B.B. Chauduri, Some shape definitions in fuzzy geometry of space, *Pattern Recognition Letters* **12**, (531–535), 1991.

[4] B.B. Chauduri, Fuzzy convex hull determination in 2D space, *Pattern Recognition Letters* **12**, (591–594), 1991.

[5] B.B. Chauduri, Concave fuzzy set: a concept complementary to the convex fuzzy set, *Pattern Recognition Letters* **13**, (103–106), 1992.

[6] B.M. Collins, Data visualization, in *Directions in Geometric Computing*, Ed. R.R. Martin, (31–80), Information Geometers, 1993.

[7] A.R. Gandhi and A. Myklebust, A natural language approach to feature based modeling, *Proc. Conf. on CAD and Computer Graphics*, International Academic Publishers, Beijing, 1989.

[8] A.R. Gandhi, *Feature Based Geometric Modeling using B-spline Surfaces with a Natural Language Approach*, Ph.D. thesis, Virginia Polytechnic Institute and State University, 1989.

[9] K.C. Gupta and S. Ray, Fuzzy plane projective geometry, *Fuzzy Sets and Systems* **54**, (191–206), 1993.

[10] C.M. Hoffmann, On the semantics of generative geometry representations, in *Design Theory and Methodology DTM 93*, Ed. T.K. Hight and L. Stauffer, ASME Vol. DE-53, 1993.

[11] G.J. Klir and T.A. Folger, *Fuzzy Sets, Uncertainty and Information*, Prentice Hall, 1988.

[12] A.D. Marshall and R.R. Martin, *Computer Vision, Models, and Inspection*, World Scientific, 1992.

[13] S.K. Pal and A. Ghosh, Fuzzy geometry in image analysis, *Fuzzy Sets and Systems* **48**, (23–40), 1992.

[14] S. Parry-Barwick and A. Bowyer, *Woodwark's method for feature recognition*, School of Mechanical Engineering, University of Bath, Technical Report 93/002, 1993.

[15] D.T. Pham and Y. Yang, A fuzzy logic approach to representing geometric uncertainties in manufacturing plans, *Proc. Applications of Artificial Intelligence in Engineering Conf.*, ISBN 1853120847, 1990.

[16] M.J. Pratt, Synthesis of an optimal approach to form feature modelling, *Proc ASME Conf. on Computers in Engineering*, San Francisco, August 1988.

[17] M.J. Pratt, Applications of feature recognition in the product lifecycle, *Int. J. Computer Integrated Manufacturing* **6**,1&2 (13–19), 1993.

[18] A.A.G. Requicha, Towards a theory of geometric tolerancing, *International Journal of Robotics Research* **2**,4 (45–60), 1983.

[19] A.A.G. Requicha and S.C. Chan, Representation of geometric features, tolerances and attributes in solid modellers based on constructive geometry, *IEEE Journal of Robotics and Automation* **RA-2**,3 (156-166), 1986.

[20] A. Ricci, Constructive geometry for computer graphics, *Computer Journal* **16**,2 (157–160), 1973.

[21] A. Rosenfeld, Fuzzy digital topology, *Information and Control* **40**, (76–87), 1979.

[22] A. Rosenfeld, On the connectivity properties of grayscale pictures, *Pattern Recognition* **16**,1 (47–50), 1983.

[23] A. Rosenfeld, The diameter of a fuzzy set, *Fuzzy Sets and Systems* **13**, (241–246), 1984.

[24] A. Rosenfeld, The fuzzy geometry of image subsets, *Pattern Recognition Letters* **2**, (311–317), 1984.

[25] A. Rosenfeld and S. Haber, The perimeter of a fuzzy set, *Pattern Recognition* **18**,2 (125–130), 1985.

[26] V. Shapiro, Real functions for representation of rigid solids, *to appear, Computer Aided Geometric Design*.

[27] D. Thalmann and N. Magenat-Thalmann, Artificial intelligence in three-dimensional computer animation, *Computer Graphics Forum* **5**, (341–348), 1986.

[28] D.G. Ullman, The evolution of function and behaviour during mechanical design, in *Design Theory and Methodology DTM 93*, Ed. T.K. Hight and L. Stauffer, ASME Vol. DE-53, (91–103), 1993.

[29] R.F. Woodbury, Variations in solids: a declarative treatment, *Computers and Graphics* **14**,2 1990.

[30] J.R. Woodwark, A dinosaur returns: metaballs, *Computer-Aided Design* **25**,11 (682–683), 1993.

[31] B. Wyvill, C. McPheeters and M. Novacek, Specifying stochastic objects in a hierarchical graphics system, *Proc. Graphics Interface '85*, (17–20), Springer Verlag, 1985.

[32] G. Wyvill, C. McPheeters and B. Wyvill, Data structure for soft objects, *The Visual Computer* **2**, (227–234), 1986.

[33] Y. Yamaguchi, H. Nakamura and F. Kimura, Probabilistic solid modelling: a new approach for handling uncertain shapes, in *Geometric Modeling for Product Realisation*, Eds. P.R. Wilson, M.J. Wozny and M.J. Pratt, (95–108), North Holland, 1992.