# Tracking topological changes in feature models

Hilderick A. van der Meiden* and Willem F. Bronsvoort†
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Mekelweg 4, NL-2628 CD Delft,The Netherlands

## Abstract

Current feature models do not explicitly represent the relation between the parameters and the topology of the model. For theoretical and practical purposes, it is important to make this relation more explicit.

A method is presented here that determines parameter values for which the topology of a feature model changes, i.e. the critical values of a given variant parameter. The considered feature model consists of a system of geometric constraints, relating parameters to feature geometry, and a cellular model. The cellular model partitions Euclidean space into quasi-disjoint cells, determined by the intersections of the feature geometry. Our method creates a new system of geometric constraints to relate the parameters of the model to topological entities in the cellular model. For each entity that is dependent on the variant parameter, degenerate cases are enforced by specific geometric constraints. Solving this system of constraints yields the critical parameter values.

Critical values can be used to compute parameter ranges corresponding to families of objects, e.g. all parameter values which correspond to models that satisfy given topological constraints.

**CR Categories:** J.6 [Computer Applications]: Computer-Aided Engineering—Computer-Aided Design I.3.5 [Computing Methodologies]: Computer Graphics—Computational Geometry and Object Modeling

**Keywords:** feature modelling, parametric modelling, families of objects, topology, constraint solving

## 1 Introduction

Feature models are based on a dual-representation scheme, consisting of a parametric representation and a geometric representation [Shapiro and Vossler 1995]. The parametric representation describes the relation between the parameters and the geometry of the features in the model. Typically the parametric representation is a CSG representation with geometric constraints. The geometric representation describes the geometry and topology of the solid shape formed by all features combined. Typically the geometric representation is a boundary representation (B-rep), generated by solving the constraints in the parametric model, and combining the features using Boolean operations, i.e. boundary evaluation.

The properties of such dual-representation schemes are not well understood, even though they are at the basis of all predominant modelling systems. To improve compatibility and reliability of feature models, CAD systems must be able to relate changes in the parametric representation to changes in the geometric representation, and vice versa.

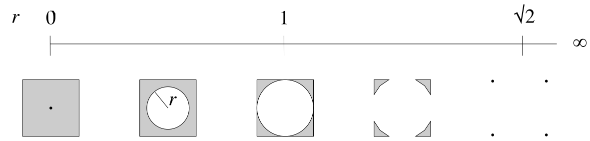



Figure 1: Critical values for a 2D model, consisting of the difference of a square with edge length 2 and a disk with radius $r$. For each critical value ($r = 0$, $r = 1$ and $r = \sqrt{2}$), the corresponding model is shown. In addition, one example model is shown for each interval between two subsequent critical values.

The geometry of feature models is declaratively specified with constraints and can be intuitively controlled via user-specified parameters. The topology of the model, on the other hand, is generated procedurally from the geometry, and cannot be controlled independently. Topology is, nevertheless, an important aspect of the design intent of a model. Therefore, the relation between the parameters and the topology of the model should be made explicit to the designer.

We present here a method to track topological changes in feature models when one parameter, called the *variant parameter*, is varied. The method determines so-called *critical values*, which are the values of the variant parameter for which the topology of the model changes. This concept is illustrated in Figure 1.

The basic approach to tracking topological changes is as follows. We create a system of geometric constraints to relate the parameters of a model to topological entities in the geometric representation. From the decomposition of this system into its rigid subsystems, we determine which entities are dependent on the variant parameter. For each dependent entity, *degenerate cases* are constructed by adding geometric constraints to the system. The degenerate cases are solved using a geometric constraint solver, and from the solutions the corresponding values of the variant parameter are computed, i.e. the critical values. This process is repeated for different topological variants of the model, until all critical values have been found.

The critical values can be used to determine a *parameter range*, i.e. all values of the variant parameter corresponding to *valid* models. One definition for a valid model is a model with the same topology as a given prototype model, i.e. all models in the same representation-space family as the prototype model are valid [Shapiro and Vossler 1995]. In a more advanced approach, valid models are defined using topological constraints [Bidarra and Bronsvoort 2000; van der Meiden and Bronsvoort 2006b]. Then the parameter range corresponds to all parameter values for which the model satisfies the topological constraints. Knowing the parameter range can help a designer pick parameter values to create variations of a model that are guaranteed to be valid.

*e-mail:H.A.vanderMeiden@tudelft.nl
†e-mail:W.F.Bronsvoort@tudelft.nl

## 2 Related work

The term dual-representation scheme was first introduced in [Shapiro and Vossler 1995] to describe current parametric and feature-based CAD models, which use a parametric representation and a geometric representation. Such models have been used in commercial CAD systems for years, even though there was no proper theoretical framework. As a result, many inconsistencies in models and incompatibilities between models of different CAD systems still exist, as documented in, for example, [Shapiro and Vossler 1995; Bidarra and Bronsvoort 2000].

The properties of dual-representation schemes have, in particular, been studied in the context of families of objects. For a dual-representation scheme, a parameter-space family and a representation-space family can be distinguished. The parameter-space family associated with a model is the set of objects corresponding to all the variants of the model with different parameter values. The representation-space family associated with a model is the set of objects corresponding to all the models that can be obtained by a continuous transformation in the geometric representation of the given model. How continuity is defined, depends on the actual representation used, but the intention is that a representation-space family represents a set of objects with similar topology.

A representation-space family typically corresponds to a subset in the parameter space, i.e. for all parameter values in this subset, the topology of the model is similar. The parameter-space family of a model then corresponds to several representation-space families, i.e. geometric representations with different topology. This relation is defined procedurally, by the algorithm a CAD system uses to generate a geometric representation from a parametric representation. To determine the subset in the parameter space that corresponds to a particular representation-space family, the inverse relation, from topological entities to parameters, must also be known.

Specific representation-space families have been defined for various types of geometric representations, e.g. in [Raghothama and Shapiro 1998; Raghothama and Shapiro 2002; Raghothama 2006]. However, in these studies the relation between topology and parameters is not considered.

All modern CAD models use geometric constraints to relate parameters to feature geometry. In [van der Meiden and Bronsvoort 2006a] a method is presented to find the range of parameter values associated with all solutions of a system of geometric constraints, for a single variant parameter. This parameter range can be used to help the user explore the parameter-space family associated with a feature model. This method does not, however, consider the topology of the model.

A particular method for computing a parameter interval corresponding to a given topology is presented in [Hoffmann and Kim 2001]. The method deals only with a very simple model: a 2D rectilinear polygon, consisting of horizontal and vertical line segments, with distance constraints between pairs of parallel lines. The method determines, for a single distance parameter, the interval such that the topology of the polygon does not change. Because of the rectilinear geometry, horizontal and vertical constraints can be considered separately, so the problem is essentially 1D.

The method presented in this paper tracks topological changes in a much more general model, and computes the corresponding critical values for a given parameter. An interval between two subsequent critical values corresponds to a representation-space family, i.e. a set of objects with similar topology.

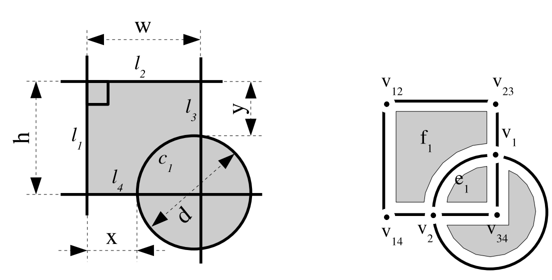
## 3 Relating parameters and topology

The feature model considered here is based on the Semantic Feature Model [Bidarra and Bronsvoort 2000]. The parametric representation is the *canonical shape model*. Geometry is defined in terms of *carriers* and *half-spaces* [Rossignac and O'Connor 1988]. A carrier is defined as the set of points $p$ that satisfy an equation of the form $f(p) = 0$. Associated with a carrier are two half-spaces, $f(p) < 0$ and $f(p) > 0$. Carriers used in our model are limited to planar, cylindrical and spherical surfaces, and the half-spaces used are those induced by such carriers. Features are parameterised volumetric shapes, defined by a Boolean combination of half-spaces. The corresponding carriers are variables in a system of geometric constraints, which determines their size and relative position and orientation. Parameters of the constraints can be used as feature parameters.

The geometric representation is the *cellular model* (CM), a cell-complex representation that can be used to store semantic feature information [Bidarra et al. 1998] and can be updated efficiently [Bidarra et al. 2005]. The cellular model represents topological entities, i.e. vertices, edges, faces and cells, and all incidence relations between these entities. Note that in literature on cell-complex representations, usually all topological entities are called cells, whereas here we use the word cell only for those entities representing volumes. All cells are quasi-disjoint, meaning that they may touch (they share a face, edge or vertex), but they cannot intersect. Each cell in the cellular model belongs to a single feature volume or the intersection of several feature volumes. Thus, each cell is completely inside or completely outside any given feature volume. The *nature* of a cell specifies whether it contains material or not.

The relation between features, carriers and topological entities is illustrated in Figure 2 for a simple 2D model. The canonical shape model in Figure 2(a) defines four linear carriers, $l_1 ... l_4$, and one circular carrier, $c_1$. The carriers are related by dimension parameters $h$, $w$, $x$, $y$ and $d$. These relations can be represented by a constraint graph, as shown in Figure 3(a).

The cellular model stores the geometry of entities and the topological relations between entities. Figure 2(b) shows such relations, e.g. vertices $v_1$ and $v_2$ are the endpoints of edge $e_1$, and edge $e_1$ is on the boundary of face $f_1$. The geometry of entities is determined by intersections of carrier geometry. For example, edge $e_1$ is determined by the intersection of the circular carrier $c_1$ with the half-spaces induced by the carriers $l_3$ and $l_4$. Vertices $v_1$ and $v_2$ are determined by intersections of carrier $c_1$ with carriers $l_3$ and $l_4$ respectively. Face




(a) canonical shape model (b) cellular model

Figure 2: (a) shows two features: a box defined by four linear carriers $l_1 ... l_4$ and a disk defined by circular carrier $c_1$. (b) shows topological entities: vertices are labelled $v_*$, edges $e_*$ and faces $f_*$

.

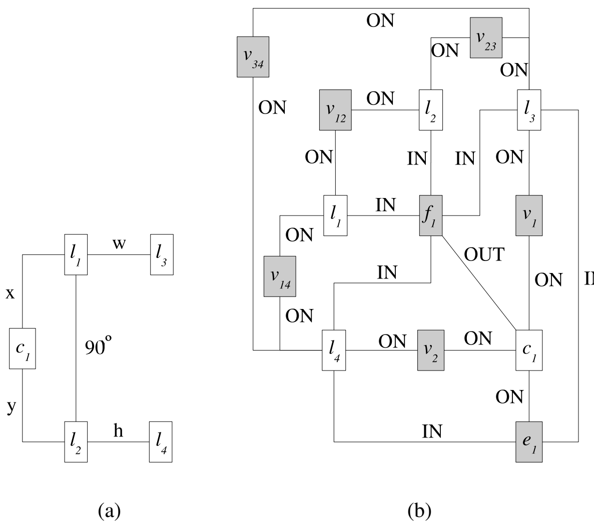



Figure 3: Constraint graphs relating (a) parameters and carriers, (b) carriers and entities. White boxes represent carriers and gray boxes represent characteristic points of entities. Edges in the graphs represent constraints, labelled with the name of a parameter or the type of the constraint.

$f_1$ is determined by the intersection of the half-spaces induced by carriers $c_1$, $l_1$, $l_2$, $l_3$ and $l_4$. Note that not all entities in Figure 2(b) have been labelled, for clarity.

Our method creates a system of geometric constraints that represents the relation between carriers and entities, as shown in Figure 3(b). For each entity in the cellular model, we add a variable *characteristic point* to the constraint system. Each characteristic point is constrained to the carriers and half-spaces that determine the corresponding entity. Constraints marked as ON, force a characteristic point to be incident with a carrier. Constraints marked as IN or OUT, constrain a characteristic point in one of the two half-spaces induced by a carrier. To be precise, a characteristic point that is constrained IN with respect to a carrier, is a point $p$ such that $f(p) < 0$, where $f$ is the carrier function. A characteristic point that is constrained OUT with respect to a carrier, is a point $p$ such that $f(p) > 0$.

The constraints in the canonical shape model (e.g. Figure 3(a)) and the system of constraints on characteristic points (e.g. Figure 3(b)) together represent the relation between the parameters of the model and topological entities.

## 4 Tracking algorithm

A critical value of the variant parameter is defined as any value $c$ for which there is an arbitrarily close value $c+\varepsilon$, $|\varepsilon| > 0$, such that the two values correspond to models with different topology. Critical values thus are characteristic for a change in the topology of a given model when the geometry is changed in a continuous way. Topology, in this context, is represented by a cellular model, consisting of entities of a given type and adjacency relations between entities. Models with different topologies are models with different cellular models.

Associated with critical values are degenerate entities. Degenerate entities are entities that should not occur in the geometric representation of a model. Examples of degenerate entities are edges of length 0, faces with area 0, and cells with volume 0. Such entities should be represented by lower-dimensional entities instead. Entities that represent disjoint point sets, or point sets that can be decomposed into point sets of different dimension, should be split into several entities, and are therefore also degenerate.

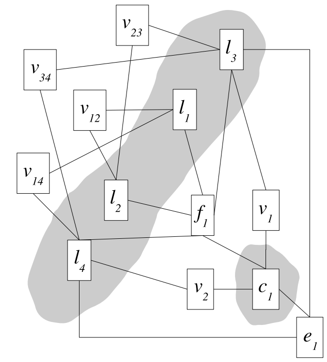



Figure 4: Root clusters, shown by the gray areas, for variant parameter $x$.

Although degenerate entities are never generated by the modelling system, we can formulate constraints on the geometry of an entity such that it would be degenerate if the constraints were satisfied. For any entity, one or more *degenerate cases* can be formulated in terms of geometric constraints. For example, a vertex, determined by the intersection of two lines, degenerates when the intersection of the two lines no longer exists. This corresponds to a constraint specifying that the lines should be parallel. All degenerate cases are discussed in Section 5.

The basic approach for tracking topological changes is as follows. In Section 3, we introduced a system of constraints relating entities to the parameters of the model. Here, we consider the variant parameter as a variable of the system, i.e. no parameter value is assigned. We then add constraints representing a degenerate case to the system. By solving the modified system, we obtain values of the variant parameter for which an entity degenerates, and thus the topology of the model must change, i.e. critical parameter values.

To be able to track all topological changes, it is essential that the topological entities of our model partition the complete Euclidean space into volumetric cells. Because space is then covered by cells, any topological change corresponds to a degenerate cell or a degenerate lower-dimensional entity on the boundary of a cell. Thus, the cellular model must also represent space outside the model with one or more cells. This may be accomplished by considering the empty space in the initial cellular model as an entity. When feature entities are added to the cellular model, this initial entity is split. Another possibility is to add a volume to the model corresponding to a bounding box or the convex hull of all feature entities.

Obviously, only entities that are dependent on the variant parameter can degenerate, i.e. the entities of which the geometry changes when the value of the variant parameter is changed. To determine whether an entity is dependent on the variant parameter, we inspect the *decomposition* of the system of geometric constraints into its rigid subsystems, called *clusters*. The decomposition determines in which order subproblems can be solved, and is one of the most important aspects of a geometric constraint solver, see [Hoffmann et al. 2001].

If we remove the constraint corresponding to the variant parameter from the model, then the system of constraints defined by the canonical shape model becomes underconstrained. In the decom-

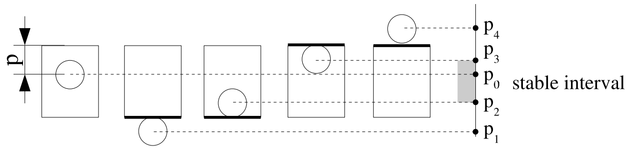



Figure 5: Stable interval for a model with a box and a circle, with a variant parameter $p$. The current parameter value is $p_0$. Four degenerate cases are found, resulting in four critical parameter values: $p_1 ... p_4$. The stable interval is $< p_2, p_3 >$.

position, the carriers that have relative degrees freedom will be in different clusters. Entities that are determined by carriers in different clusters are dependent on the variant parameter.

Figure 4 shows the decomposition of the system of constraints on carriers from Figure 3(a), given that parameter $x$ is the variant parameter, combined with the characteristic points from Figure 3(b). Here, $c_1$ is a cluster, and $l_1$, $l_2$, $l_3$, and $l_4$ form a cluster. Entities $v_{12}$, $v_{23}$, $v_{34}$ and $v_{14}$ are dependent on only one cluster. Entities $v_1$, $v_2$, $e_1$ and $f_1$ are dependent on both clusters, and therefore these entities are dependent on the variant parameter.

The algorithm for tracking topological changes works as follows. For some value of the variant parameter, the *current parameter value*, it generates the cellular model and the system of constraints, relating entities to parameters. Next, it determines which entities are dependent on the variant parameter, and it solves all degenerate cases for these entities. The algorithm then determines the interval for which the topology of the current cellular model does not change, called a *stable interval*. This is the interval between the largest of the critical values smaller than the current parameter value, and the smallest critical value larger than the current parameter value (see Figure 5).

The cellular model, when generated for different parameter values, may contain different entities, and therefore, different critical values may be found. Thus, to obtain all critical values, the cellular model must be regenerated for several values of the variant parameter, such that degenerate cases can be solved for all possible entities. The tracking algorithm picks a new current parameter value arbitrarily, but outside any previously determined stable interval. A new cellular model is generated for this parameter value, and critical values are determined as before. This process is repeated until the domain of the variant parameter is completely covered by stable intervals. By definition, new critical values are never found in any previously determined stable interval. Thus, when the whole parameter domain is covered by stable intervals, all critical values have been found.

## 5 Degenerate cases

An entity of a particular type, i.e. vertex, edge, face or cell, degenerates when its characteristic point no longer represents a point set that is valid for that type of entity. For any entity that is dependent on the variant parameter, one or more degenerate cases can be formulated in terms of constraints on its carriers.

Figure 6 shows an example of a degenerate case. An edge is determined by the intersection of a circular carrier and a half-plane defined by a linear carrier. In general, the intersection is a curve, or does not exist. There is no intersection such that the intersection is a point, because the characteristic point may not be ON the linear carrier. If we add constraints to the system such that the carriers are tangent, then the characteristic point does not exist, i.e. the entity is degenerate.

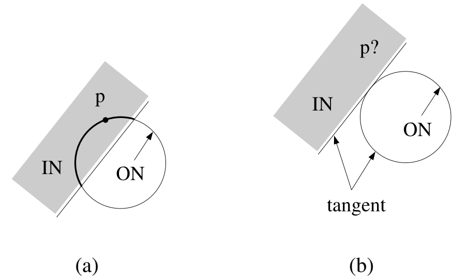



Figure 6: (a) shows an edge determined by the intersection of a circular carrier and a half-plane defined by a linear carrier. A characteristic point $p$ may be located anywhere on the edge. (b) shows that the edge degenerates when the carriers are tangent. There is no characteristic point $p$ such that it is ON the circular carrier and IN the linear half-plane.

The degenerate cases of an entity are determined using the decomposition of the system of constraints on its carriers. For each pair of carriers in different clusters, i.e. each pair of carriers with relative degrees of freedom, we add constraints such that (a) the system is well-constrained and (b) the entity is degenerate, i.e. the characteristic point of the entity no longer represents a valid entity of the given type.

Table 1 lists all possible topologically different intersections of two carriers, where a carrier is either a plane, sphere or cylinder. Each intersection corresponds to a combination of constraints imposed on the carriers. For a given pair of carriers, we determine the type of intersection by verifying the constraints in the table. Next, for all other intersections listed in the table, we determine if adding the given constraints makes the system of constraints well-constrained. If this is the case, then for each solution of the system we determine whether the entity is degenerate.

Consider, for example, a face that is constrained ON a spherical carrier and IN a cylindrical carrier. The variant parameter, $d$, is the distance of the centre of the sphere to the axis of the cylinder. Given are the radius of the cylinder, $r_c$, and the radius of the sphere, $r_s$, and in this example $r_c < r_s$. The intersection curve of a sphere and a cylinder is known as Viviani's curve [Weisstein 2006]. Suppose that in the current configuration, the intersection of the carriers is a closed curve, represented by a single edge in the cellular model. From Table 1 we can infer the constraints to force the carriers into different configurations. We find that the face degenerates to a point for $d = r_s + r_c$ and to a 'figure 8' for $d = r_s - r_c$.

Note that the system of constraints representing a degenerate case may have several solutions. If an entity is determined only by carriers, i.e. if it is constrained only ON its carriers, then, for all solutions of the system, the characteristic point represents an entity corresponding to an intersection chosen from Table 1, which is different from the current intersection. Thus, the entity degenerates for all solutions. If, however, an entity is determined by one or more half-spaces, i.e. if it is constrained IN or OUT with respect to one or more carriers, then each solution may yield a characteristic point representing a different type of entity, and we should verify for each solution whether the entity is degenerate.

For example, Figure 7 shows the topological variants of a disk and a half-plane defined by a linear carrier. The parameter $P$ is the distance from the centre of the circular carrier to the linear carrier. In this example there are five different topological variants: the disk is completely outside the linear half-plane (zero intersections), the

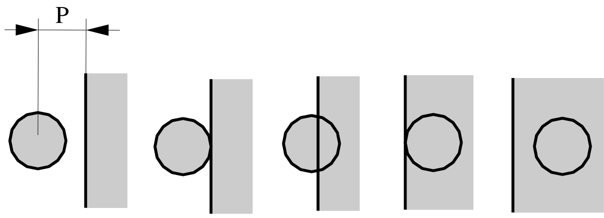



Figure 7: Five different configurations of a half-plane and a disk correspond to three different configurations of the corresponding carriers, i.e. the carriers have either zero, one or two intersection points.

disk is outside the linear half-plane and touching the line (one intersection), the disk is partially overlapping the linear half-plane (two intersections), the disk is inside the linear half-plane and touching the line (one intersection), or the disk is completely inside the linear half-plane (zero intersections). On the other hand, there are only three configurations of the carriers, i.e. such that the carriers intersect in zero, one or two points. The degenerate case of the carrier intersection, i.e. when there is one intersection point, corresponds to two configurations of the half-spaces, i.e. the disk touches the line, and is either inside or outside the linear half-plane. The first case does not correspond to a degenerate case of the intersection, the second one does.

For solving the systems of constraints corresponding to degenerate cases, a 3D constraint solver is needed that supports the constraints listed in Table 1. Also, the solver must support the constraints in the canonical shape model, such as distances and angles on planes, cylinders and spheres. Such constraint systems can be solved using, for example, [Oung et al. 2001]. In [van der Meiden and Bronsvoort 2005] it is suggested that constraint on carriers can be mapped to systems of constraints on point variables.

## 6 Parameter range computation

The range of a given parameter is the set of values corresponding to valid models. One possible definition for a valid model is a model with the same topology as a given prototype model [Shapiro and Vossler 1995]. Then the parameter range is equivalent to the stable interval, as computed in Section 4.

However, assuming that only a single representation-space family is of interest, is rather limiting from a designer's perspective. A designer is not interested in one topological variant of the model per se, but rather wants the topology of the model to correspond with his or her design intent.

Topological aspects of the design intent can be expressed by topological constraints, as is done in the Semantic Feature Modelling approach [Bidarra and Bronsvoort 2000], and The Declarative Family of Objects Model [van der Meiden and Bronsvoort 2006b]. For example, boundary constraints can specify that certain feature faces must be on the boundary of the model, or may not be on the boundary of the model. Interaction constraints can specify that certain interactions of features are not allowed. The designer is interested in the family of all objects that satisfy these constraints, referred to here as a *semantic family*.

For a given variant parameter, the parameter range corresponding to the semantic family of a model can be determined from the critical values of the model, as follows. Each interval between two subsequent critical values represents a representation-space family. We generate an object in each interval, and test whether it satisfies the topological constraints on the model. If so, then the whole interval is part of the parameter range, because all objects in the interval have the same topology, and thus satisfy the topological constraints. Each interval between two subsequent critical values can thus be marked as part of the parameter range, or not part of the parameter range.

| **intersection** | **constraints** |
|---|---|
| **plane-plane** | |
| none | parallel, $d \neq 0$ |
| line | not parallel |
| coincident | parallel, $d = 0$ |
| **plane-sphere** | |
| none | $d > r_2$ |
| point | $d = r_2$ |
| circle | $d < r_2$ |
| **plane-cylinder** | |
| none | parallel, $d > r_2$ |
| line | parallel, $d = r_2$ |
| ellipse | not parallel |
| two lines | parallel, $d < r_2$ |
| **sphere-sphere** | |
| none | $d > r_1 + r_2$ |
| point | $d = r_1 + r_2$ |
| circle | $d < r_1 + r_2$ |
| coincident | $d = 0$, $r_1 = r_2$ |
| **sphere-cylinder** | |
| none | $d > r_1 + r_2$ |
| point | $d = r_1 + r_2$ |
| closed curve | $r_{max} - r_{min} < d < r_1 + r_2$ |
| 'figure 8' | $d = r_1 - r_2$, $r_1 > r_2$ |
| 2× closed curve | $d < r_1 - r_2$, $r_1 > r_2$ |
| point | $d = r_2 - r_1$, $r_2 > r_1$ |
| none | $d < r_2 - r_1$, $r_2 > r_1$ |
| **cylinder-cylinder** | |
| none | parallel, $d < r_{max} - r_{min}$ |
| line | parallel, $d = r_{max} - r_{min}$ |
| two lines | parallel, $r_{max} - r_{min} < d < r_1 + r_2$ |
| line | parallel, $d = r_1 + r_2$ |
| coincident | parallel, $d = 0$, $r_1 = r_2$ |
| none | $d > r_1 + r_2$ |
| point | $d = r_1 + r_2$ |
| closed curve | $r_{max} - r_{min} < d < r_1 + r_2$ |
| 'figure 8' | $d = r_{max} - r_{min}$ |
| 2× closed curve | $d < r_{max} - r_{min}$ |
| 2× 'figure 8' | $d = 0$, $r_1 = r_2$ |

Table 1: Intersections of pairs of carriers and corresponding constraints. Legend: $r_1$ = the radius of carrier 1, if it is a sphere or cylinder. $r_2$ = the radius of carrier 2, if it is a sphere or cylinder. $r_{min} = min(r_1, r_2)$. $r_{max} = max(r_1, r_2)$. $d$ = distance between any two of: a plane, the centre of a sphere or the axis of a cylinder.

Knowing the parameter range can help a designer to maintain model validity while changing a parameter of the model. There will be no undesirable changes in the semantics of the model if the parameter value is restricted to the parameter range. This can be very helpful when fine-tuning a model. Also, presenting one or more parameter ranges to the designer may help him/her understand the model better, in terms of the family of objects that it represents. Although the relation between the ranges of different parameters cannot be explicitly represented, it can be explored by varying one parameter at a time.

## 7 Conclusions

We have presented a method to compute the critical values when a single parameter of a feature model is varied, i.e. the parameter values for which the topology of the model changes. Our method consists of two parts: the construction of a system of constraints that relates the parameters of a model to topological entities, and an algorithm to track topological changes, using that system of constraints.

The method can be used to compute the parameter range corresponding to valid models, i.e. a single representation-space family or a semantic family. Knowing this range can help designers to create only objects with the correct semantics.

Our approach of relating parameters and topological entities using a system of geometric constraints, can also be used for other representations than those used here, e.g. the commonly used combination of a CSG-like representation with a B-rep. The algorithm for tracking topological constraints, however, relies on a cellular model. It remains to be seen whether a similar tracking algorithm can be devised for B-reps and other geometric representations.

The method presented here can only track topological changes for one parameter at a time. Computing the parameter ranges when two parameters, or perhaps three parameters, are varied, may also be useful for a designer. Considering even more parameters is not useful for a designer, since the high-dimensional parameter space is difficult to interpret [Hoffmann and Kim 2001]. However, simultaneously considering more parameters may be useful for automated model optimisations.

The carrier geometry considered here is limited to planes, cylinders and spheres. For many applications, more general algebraic geometry and parametric geometry such as NURBS should also be considered. For this, a more generic approach to formulating and solving degenerate cases will be needed.

The presented method for tracking topological changes was motivated by important problems in CAD, but it may also be useful in other application where geometric constraints and representations are used, e.g. CAM and robotics.

## Acknowledgements

H.A. van der Meiden's work is supported by the Netherlands Organisation for Scientific Research (NWO).

## References

BIDARRA, R., AND BRONSVOORT, W. F. 2000. Semantic feature modelling. *Computer-Aided Design 32*, 3, 201–225.

BIDARRA, R., DE KRAKER, K. J., AND BRONSVOORT, W. F. 1998. Representation and management of feature information in a cellular model. *Computer-Aided Design 30*, 4, 301–313.

BIDARRA, R., MADEIRA, J., NEELS, W., AND BRONSVOORT, W. F. 2005. Efficiency of boundary evaluation for a cellular model. *Computer-Aided Design 37*, 12, 1266–1284.

HOFFMANN, C. M., AND KIM, K.-J. 2001. Towards valid parametric CAD models. *Computer-Aided Design 33*, 1, 81–90.

HOFFMANN, C. M., LOMONOSOV, A., AND SITHARAM, M. 2001. Decomposition plans for geometric constraint systems, Part I: performance measures for CAD. *Journal of Symbolic Computation 31*, 4, 376–408.

OUNG, J., SITHARAM, M., MORO, B., AND ARBREE, A. 2001. FRONTIER: fully enabling geometric constraints for feature-based modeling and assembly. In *Proceedings of Solid Modeling '01, Sixth ACM Symposium on Solid Modeling and Applications, June 4-8, Ann Arbor, USA*, ACM Press, D. C. Anderson and K. Lee, Eds., 307–308.

RAGHOTHAMA, S., AND SHAPIRO, V. 1998. Boundary representation deformation in parametric solid modeling. *ACM Transactions on Graphics 17*, 4, 259–286.

RAGHOTHAMA, S., AND SHAPIRO, V. 2002. Topological framework for part families. *Journal of Computing and Information Science in Engineering 2*, 4, 246–255.

RAGHOTHAMA, S. 2006. Constructive topological representations. In *Proceedings ACM Symposium on Solid and Physical Modeling, Cardiff, Wales, United Kingdom, June 6-8*, ACM SIGGRAPH, L. Kobbelt and W. Wang, Eds., 39–51.

ROSSIGNAC, J. R., AND O'CONNOR, M. A. 1988. SGC: a dimension-independent model for pointsets with internal structures and incomplete boundaries. In *Proceedings of the 1988 IFIP/NSF Workshop on Geometric Modeling, Renselaerville, New York, USA*, M. Wozny, J. Turner, and K. Preiss, Eds., 145–180.

SHAPIRO, V., AND VOSSLER, D. L. 1995. What is a parametric family of solids? In *Proceedings of the Third ACM/IEEE Symposium on Solid Modeling and Applications, May 17-19, Salt Lake City, Utah, USA*, ACM Press, C. M. Hoffmann and J. R. Rossignac, Eds., 43–54.

VAN DER MEIDEN, H. A., AND BRONSVOORT, W. F. 2005. An efficient method to determine the intended solution for a system of geometric constraints. *International Journal of Computational Geometry and Applications 15*, 3, 279–298.

VAN DER MEIDEN, H. A., AND BRONSVOORT, W. F. 2006a. A constructive approach to determine parameter ranges for systems of geometric constraints. *Computer-Aided Design 38*, 4, 275–283.

VAN DER MEIDEN, H. A., AND BRONSVOORT, W. F. 2006b. Solving topological constraints for declarative families of objects. In *Proceedings ACM Symposium on Solid and Physical Modeling, Cardiff, Wales, United Kingdom, June 6-8*, ACM SIGGRAPH, L. Kobbelt and W. Wang, Eds., 63–71.

WEISSTEIN, E. W., 2006. Viviani's curve. From MathWorld–A Wolfram Web Resource.
http://mathworld.wolfram.com/VivianisCurve.html.