BOUNDARY REPRESENTATION MODELLING WITH LOCAL TOLERANCES

David J. Jackson

Parasolid Business Unit,
EDS Unigraphics,
Parker's House, 46 Regent Street,
Cambridge, England.

ABSTRACT

Conventional boundary representation (b-rep) solid modellers perform all computations to a single global tolerance. For solids bounded by freeform surfaces, this means that an excessive amount of data may be required to define the surface to the required accuracy. Given that different modellers invariably operate to different modelling tolerances, global tolerances are also a major roadblock to the robust exchange of b-rep models between systems.

This paper describes an extension to b-rep modelling to allow different features in a model to be constructed to different tolerances. This approach, which we call 'tolerant modelling', has been implemented and released in a commercially used solid modelling kernel - EDS' Parasolid modeller.

A tolerance is associated with each face, edge and vertex of the b-rep; subsequent modelling operations take account of these local tolerances. This enables the efficient use of construction techniques which can operate at a tolerance appropriate to the particular feature. Importing, combining, and modelling with features created in several different surface and solid modelling systems also becomes possible.

This paper discusses the problems associated with traditional approaches to handling tolerances in b-rep modellers. The principles of tolerant modelling are then introduced, and a boolean algorithm that operates on tolerant models is described. Finally, aspects of other modelling operations and data exchange are discussed.

1. INTRODUCTION

This paper describes a type of boundary representation which has recently been implemented in the Parasolid solid modeller, and which allows the modeller to represent and utilise geometry regardless of the tolerance to which it was created, and to mix geometries of different tolerances in the same model.

As development of solid modellers has progressed, much work has been done in an effort to make modelling operations, and booleans in particular, reliable. Ideally, it should be possible to take any two solids which the modeller regards as valid, and unite them into a valid solid, which correctly models the set-theoretic union (to an appropriate tolerance).

In practise modellers only approach this level of reliability, especially when dealing with free-form surfaces. Not only is it difficult to design reliable algorithms to intersect curves and surfaces, it is even harder to incorporate these into a boolean algorithm which deals consistently with the results of the many intersection and coincidence enquiries which take place in a complex boolean operation.

Tolerant modelling is a foundation on which reliable solid modelling can be built. By attaching local tolerances to faces, edges and vertices, the boolean algorithm can perform intersections and coincidence tests to appropriate tolerances, and take advantage of a richer structure to represent the resulting solid.

2. BACKGROUND

2.1 Other approaches to robustness

A number of different approaches to the problem of robust geometric modelling have been put forward.

Exact (rational) arithmetic has been used [1], but this suffers from the build-up of complexity of the numerical values (mitigated somewhat by only evaluating the exact values when required). Moreover, it may create small undesirable features in the model which do not correspond to any design intent.

Potentially ambiguous configurations can be avoided by applying small perturbations to the geometry before the boolean [2]. Again, this creates small edges and faces, and removes coincidences which were probably intentional.

Symbolic reasoning systems have been used to ensure that certain consistency statements about the various incidence or coincidence relations between the geometries remain true. In other words if certain relations are computed, others can be deduced. The difficulty here is describing all the consistency requirements in the general case; so far this is limited to special cases [3], [4].

In [5], the authors describe an algorithm which uses a hybrid CSG/b-rep model to perform boolean operations by avoiding redundancy when evaluating the boundary of a manifold CSG model. A global tolerance is used.

Solid Modeling '95, Salt Lake City, Utah USA

Adaptive tolerance-based approaches associate tolerances with individual points, curves and surfaces, and detect ambiguous situations. When an ambiguity is detected, the tolerances have to be increased, and some or all of the algorithm re-run. [6] uses a CSG/b-rep model, whereas [7] is limited to polyhedral b-reps.

2.2 *Use of tolerances*

Most geometric modelling systems use small tolerances when determining answers to questions of intersection or coincidence. Two points are considered coincident if their distance is less than the tolerance value; two edges are coincident if they are everywhere within tolerance of each other, and so on.

The tolerance also determines the accuracy to which computations are performed, for example when converging to an intersection between two curves.

Tolerances help in several ways:

- By setting the tolerance several orders of magnitude above the floating point precision available, the modeller can be largely unaffected by rounding error.
- The tolerance prevents the creation of very small faces and edges which would serve no practical purpose in the representation of the model, and cause problems for downstream applications.
- Consistent use of tolerances to guide decisions in modelling operations can avoid inconsistent decisions leading to conflict between topology and geometry.

Some modellers use ad-hoc tolerances, but most use a single global distance which can be set by the user.

2.3 *Limitations of a single tolerance*

The approach taken by many modellers, and by Parasolid until recently, was to choose a fixed global tolerance value, and to recommend that all systems using the modeller use the same tolerance, so that models can be exchanged between them. In Parasolid's case this value was 5.0e-9 metres, chosen to be below the smallest realistic tolerance for most mechanical parts.

The difficulty is that there is no single tolerance which will be suitable for all requirements. A very small value will necessitate the creation of surfaces with a large amount of defining data in operations such as blending, but a large tolerance on a model prevents the creation of small features anywhere on that model.

Moreover, if trimmed surfaces are imported from another modelling system, they will only meet to that system's tolerance. In some cases the surfaces can be extended and re-intersected to impose any desired accuracy, but in the case of surfaces which meet smoothly this may be impossible (Figure 1). It may be undesirable or impractical to modify the surfaces to meet more accurately. Consequently the global tolerance would have to be increased to encompass the worst case.

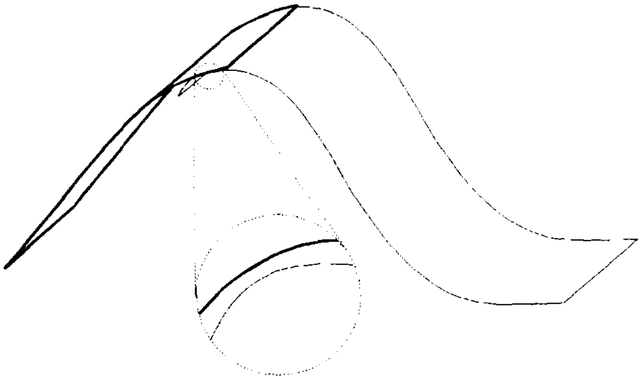

Figure 1. Tangential surfaces may fail to meet at a higher resolution

A larger fixed tolerance is also more likely to run into consistency problems - for example, it is not possible to split a short edge, of length less than 4 times the tolerance, without creating edges which are too short, or edges which do not meet up. Although short edges like this would not normally be modelled deliberately, they can easily occur in the course of modelling a complex part. Figure 2 shows an example of this, where a face with a short edge is being split by the vertical dashed line. The modelling tolerance is represented by dashed tolerance regions around the edges and vertices. The vertex which would be created in the middle of the short edge would be coincident with both ends of the edge.

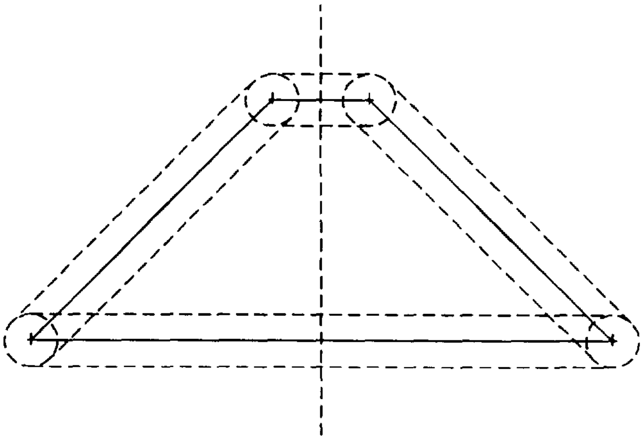

Figure 2. Dividing a face, through a short edge.

2.4 *Tolerant Modelling*

In tolerant modelling, a local tolerance is associated with every face, edge and vertex in the model. This provides flexibility in representing geometric models which enables straightforward transfer of models between systems, and a foundation for robust solid modelling.

When necessary, tolerances are locally increased and topology merged to avoid ambiguity.

3. OVERVIEW OF TOLERANT MODELLING

3.1 Model Representation

Each vertex has an associated tolerance, which is a distance value. This can be the 'default' tolerance (normally 5.0e-9 metres) or any larger value. The tolerance region around the vertex is a sphere, of radius equal to the tolerance, centred at the point of the vertex.

An edge also has an associated tolerance. If this is greater than the default, the edge is called a 'tolerant' edge. An edge with default tolerance is represented by a single 3-space curve. A tolerant edge is represented by a collection of curves, in the parameter spaces of the faces connected to it. (For a manifold solid edge, there will be two such curves). The tolerance region around the edge is a tube of radius equal to the edge tolerance, centred on one of the parameter space curves. The choice of curve is arbitrary, but fixed for each edge. Thus all the parameter-space curves lie in this tube. Figure 3 shows a face with two tolerant edges, the tolerance regions are shown dotted.

Faces also have associated tolerances, which define the 'thickness' of the face. In Parasolid faces currently always have the default tolerance.

The vertex tolerance must be at least as great as that of connected edges, and the edge tolerance must be at least as great as that of connected faces.

3.2 Model Consistency

Intersection and coincidence tests are always performed relative to appropriate tolerances - e.g. two entities intersect if they approach within the sum of their tolerances.

With this interpretation, the usual b-rep consistency rules apply - vertices must lie on connected edges and faces, edges must lie on connected faces, and edges and faces must only intersect where topology exists.

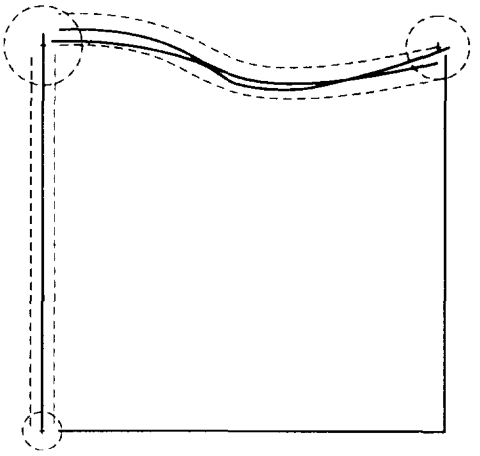

Figure 3. Parameter-space curves lie in tolerance regions

In particular, the ends of the parameter-space curves on an edge must lie inside the appropriate vertex sphere. Note however that the curves bounding a face need not meet exactly end-to-end, as long as they are within vertex tolerance.

More precisely, edges must only intersect at their end vertices. If the intersection of the tolerance tubes of two edges is non-empty, then each connected component of that intersection must intersect a common vertex of the two edges. Figure 4 shows an invalid face, where two edges intersect in their interior, and a valid face, where the intersection area is contractible onto the vertex.

Faces must only intersect at vertices or along edges. More precisely, if the intersection of the tolerance regions of two faces is non-empty, then each connected component of that intersection must be contractible onto a collection of common edges or

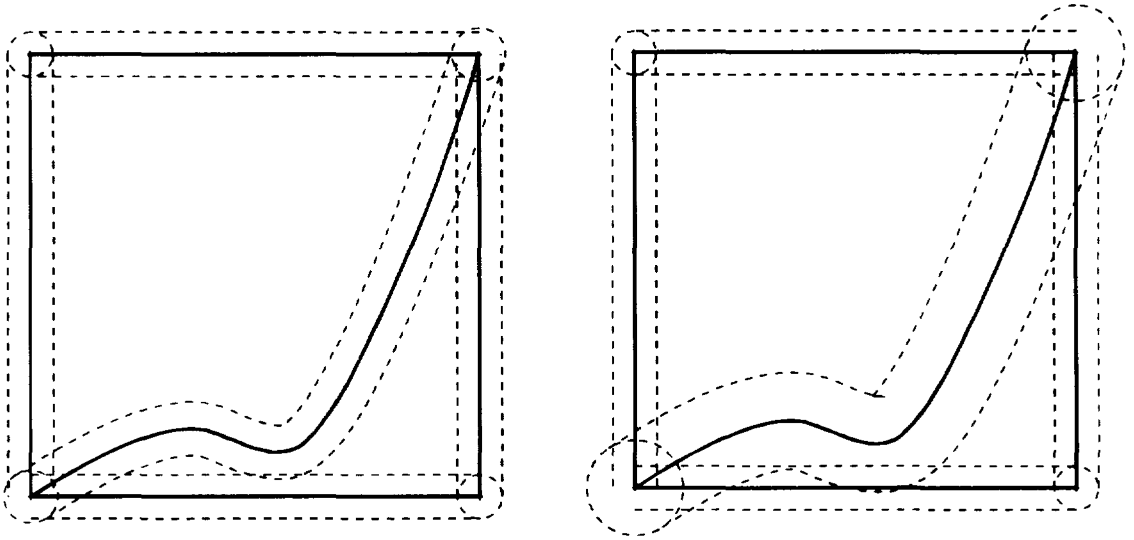

Figure 4. The left face is invalid (self-intersecting). The right face is valid as the intersection is contractible to a vertex.

vertices.

3.3 *General Principles*

There are three main reasons for the introduction of local tolerances:

1. To allow the importing of b-rep or surface models built in other systems, whatever their accuracy.
2. To enable the use of modelling techniques such as blending which generate approximated surfaces or curves, without compromising the accuracy of the rest of the model.
3. By identifying areas which have been approximated, to avoid performing numerical computations to inappropriately high accuracy, which would be slow and unreliable.

Note that tolerances are associated with topology, not geometry. Points, curves and surfaces are treated as 'exact', as are derived objects such as point-on-surface (i.e. u,v, point, surface) or trimmed curves. That is, they have the 'default' tolerance.

When a geometric test such as coincidence or intersection is performed, a tolerance equal to the sum of the tolerances of the topologies is used. When a topological operation such as splitting an edge or adding a face cannot be performed with existing tolerances, because entities collide, then the local tolerances are increased, and redundant topology removed, until a consistent model can be created.

4. BOOLEAN OPERATIONS

4.1 *Introduction*

This section describes a boolean algorithm which can be used on models with local tolerances. It applies to conventional solid b-reps, or general mixed-dimension cellular non-manifold models.

The boolean operations of unite, subtract and intersect operate on two models and perform an approximation to the corresponding set-theoretic operations. In a robust modeller, the result of a unite will always be a valid b-rep which models the set-theoretic union to the appropriate tolerances.

4.2 *Phases of the boolean*

The boolean algorithm described here proceeds in the usual three phases which we call imprint, join and select.

Imprint

Initially, the two models may have topologies which partially overlap each other, e.g. crossing edges, crossing faces, or vertices in the middle of edges. The purpose of the imprint phase is to add topology, to divide existing faces and edges so that each topological entity in one model intersects entities in the other model only at other topologies. The interpretation here is the same as if the two had been part of the same model.

The result of the imprint is a list of topologies on one body, and matching topologies on the other. Note that a vertex can correspond to an edge, face or even a 3-space region (cell), if the tolerances so indicate.

Adding topology in this way can require additional changes to either body to maintain a valid b-rep. For example, an edge may be split to create a new vertex, which may be within tolerance of another edge. This may require the removal of a thin face which has degenerated to an edge. Figure 5 shows an example of this - when the faces on the left are split by imprinting the vertical line, two of the new vertices are coincident. These vertices need to be combined, and the thin degenerate face removed, as shown on the right.

Join

This phase performs two functions - the compression of topologies on the input models, and the actual combination into one model, which will in general be non-manifold and composed of a number of regions or cells bounded by faces.

For example if a vertex on one model coincides with an edge in the other, so that the edge is entirely contained within the tolerance sphere of the vertex, then that edge needs to be removed, and its ends combined, before the models can be joined.

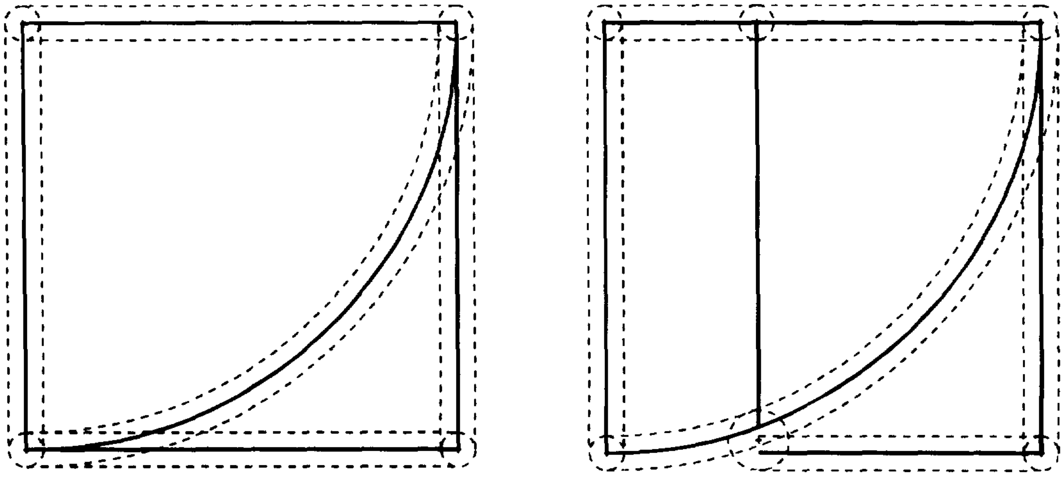

Figure 5. When the vertical line is imprinted, a thin face needs to be removed.

Select

Depending on the operation (unite, intersect, subtract) and on the type of 'regularisation' required, some of the merged model will survive, the rest is deleted. The selection phase determines which entities are kept, based on inside/outside classification, which is possible because regions must be unambiguously inside or outside after the imprint phase.

4.3 Imprint

The algorithm compares low-dimension topologies of the two models, before moving on to higher ones. This simplifies the capture of all geometric relationships.

a) vertex-vertex.

Determine coincident vertices. Note that one vertex may correspond to more than one on the other model.

b) vertex-edge.

If a vertex lies inside an edge, split the edge and make the two correspond. Note that splitting the edge may result in removal of nearby topology, some of which may already have been matched to topology in the other model. The new vertex inherits the tolerance of the split edge.

c) edge-edge.

Intersect pairs of edges, using the sum of the edge tolerances. The edges will intersect in 'areas of coincidence' which are parametric ranges within each edge. Intersections which represent an area of coincidence which contains a vertex can be ignored, as they will have been found by vertex-edge comparisons. Any other intersections will result in splitting both edges.

The requirements on the curve intersector are that an intersection is returned for each area of coincidence, to tolerance, between the two curves.

d) vertex-face

If a vertex lies in the interior of a face (not at a vertex or on an edge), add a corresponding vertex to the face.

e) edge-face

Intersect the curve of the edge with the surface of face, to get intervals of coincidence on the curve. Since the edge has already been compared with edges of the face, we need only consider coincidence, or intersections interior to the face and edge.

For coincidence, add an edge to the face (between two existing vertices), and make it correspond to the edge.

For intersections, add vertices to the face and edge.

If an edge is split by this process, the pieces need to be re-compared with other edges, since the smaller edges may now be coincident when they were not before.

f) face-face

Intersect the surfaces of the faces. The resulting curves can then be compared with each face. The surface intersector needs to return curves and point contacts, sufficient to divide areas of surface which are on one side of the other surface from those on the other side; and to represent each area of contact or coincidence. It need not consider curves (i.e. contact areas) already known to be common by virtue of edge-edge or edge-face comparison. This can be important, since computation of near-tangent intersection curves can be very unstable.

Trim the curves of intersection against the boundaries of the faces. Sections of curve which appear on both faces are added as new (corresponding) edges.

Again, any split edges need to be re-compared for coincidence with existing edges and faces. This can be done reasonably efficiently since only edges with known common vertices need be considered.

At the end of the imprinting phase, the boundary of each model has been divided up so that each face lies inside, outside, or on, the boundary of the other model.

4.4 Join

When entities of different tolerances correspond, the join operation will use the larger of the two tolerances. This may result in removal of adjacent topology.

Topology on both models is compressed until there is a one-to-one correspondence between matched vertices, edges and faces on the two models. The two models are then topologically joined, and the resulting regions determined.

4.5 Select

After the two models have been combined into one, the regions, faces, edges and vertices which are to survive are determined. This will depend on the operation being performed, and on various other options. For example, the boolean can optionally remove lower-dimension topology, or split up a disconnected model into connected components, or return a collection of manifold sub-models where possible.

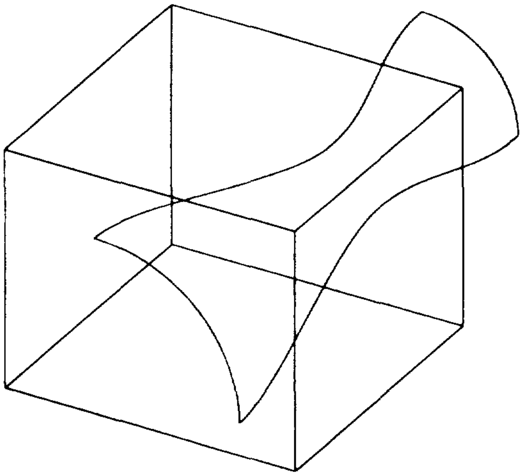

Figure 6. Subtracting a b-spline face from a block.

4.6 *Example boolean*

Figure 6 shows an example boolean operation where a 2-dimensional model is being subtracted from a block to create a blend. The face being subtracted lies on a b-spline surface, and its edges approximately coincide with two of the faces of the block. The boundary edges of this face are tolerant.

The edge/edge comparison in the imprint stage will split the long edges of the b-spline face into three. The edge/face comparison will detect that two of these edges are coincident, within tolerance, with faces of the block. New edges will be added to these faces, dividing them in two.

The face/face comparison will intersect the b-spline face with the end faces of the block, creating the end faces of the resulting blend.

The join and select phases will result in the block being split into two solids.

5. BLENDING

When a blend is created, a tolerance is supplied which is used when constructing the blend surface, and which determines how closely it fits the neighbouring faces. The edges of the blend are given a tolerance determined from this value, without affecting tolerances elsewhere in the model.

The blend is added to the model by constructing the blend face, and sewing it in along the tolerant edges.

Figure 7 shows a variable-radius blend on the edge of a cube. The tolerances are exaggerated so that the two parameter-space curves can be seen on each tolerant edge. Normally when the object is drawn only one of these curves would be shown.

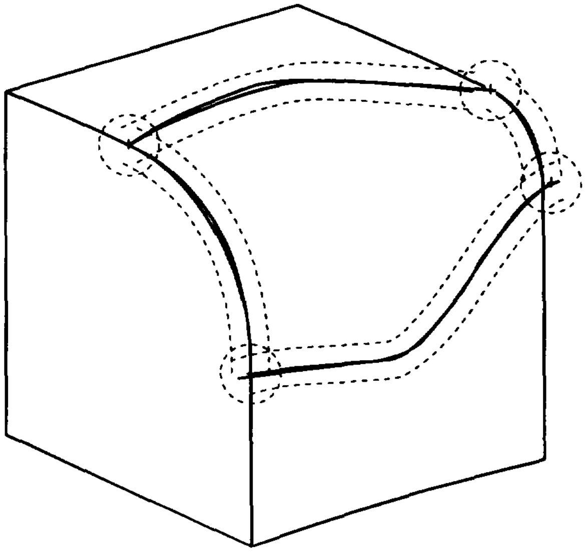

Figure 7. A blend showing exaggeratedly large local tolerances.

6. DATA EXCHANGE

Without information on the accuracy of the model, exchanging b-reps between systems cannot be performed reliably. If accuracy is only given as a single value, models can only be imported by compromising the accuracy of existing data. Tolerant modelling provides a way round this. Data from surface modellers can be imported as trimmed surfaces, for example via IGES [8], and sewn together. B-rep models can be imported directly via standards such as STEP [9].

6.1 *Sewing*

Sewing is a way of joining models along coincident edges, and is often used to import trimmed surfaces from other systems. The models to be sewn may be two-dimensional 'sheets' of faces, or solids. If the faces enclose a volume, a solid is created.

This is in fact the part of the boolean algorithm which deals with matching edges and vertices; the face-face and edge-face comparisons are omitted. This provides a faster way of combining models when they are to known to abut. As before, the edges are compared to within sum of tolerances.

If two faces do not in fact meet to the expected tolerance, a model may be created with a thin hole in it. The hole can be closed by assigning larger tolerances to the edges and re-sewing.

6.2 *STEP*

STEP is an emerging standard for the exchange of product model data. Entities added to the standard recently allow 'uncertainty' values to be associated with geometric definitions.

Solutions are currently being developed which combine the STEP framework with tolerant modelling so that local tolerance information can be attached to topology in the STEP file.

If geometry types not supported by STEP need to be transferred, they can be approximated providing the approximation tolerance is attached to the relevant faces or edges.

7. FUTURE WORK

7.1 *Face tolerances*

Currently face tolerances always take the default value in Parasolid. Face tolerances are not needed in order to define a valid b-rep from trimmed surface data, but they are useful to avoid expensive and unreliable surface intersections.

For example, if two surfaces on two models being united are approximately coincident, or approximately tangent, then intersecting them to an unsuitable tolerance will be very difficult as many complex curves may result. Marking one or both of them with a tolerance would overcome this problem.

7.2 *General compression of topology*

The recursive compression of topology resulting from features with different tolerances being matched has not been fully implemented, but a subset sufficient for practical use has been developed. Further work is required to develop a general algorithm.

7.3 Preventing tolerance growth

Compression normally results in some growth of local tolerances. Ideally models defined to a certain tolerance should unite to create models of the same or similar tolerance. Although this seems not to be achievable in all cases, tolerance growth should be contained where possible.

8. CONCLUSION

Local tolerances on a b-rep model can be used as a foundation for robust solid modelling and reliable model transfer between different systems. This technique has been implemented in a widely used production solid modeller, and has been used to solve real modelling problems.

REFERENCES

[1] Benouamer, M. O., Michelucci, D., and Peroche, B. 'Error-free boundary evaluation based on a lazy rational arithmetic: a detailed implementation'. *Computer-Aided Design,* Vol 26 No 6 (June 1994).

[2] Edelsbrunner, H. and Mucke, E. 'Simulation of simplicity: A technique to cope with degenerate cases in geometric algorithms'. *Proc. of 4th ACM symposium on Comp. Geometry* (June 1988).

[3] Hoffman, C. M., Hopcroft, J. E., and Karasick, M. S. 'Robust set operations on polyhedral solids'. *IEEE Computer Graphics and Applications* 9 (November 1989).

[4] Stewart, A.J. 'Local robustness and its application to polyhedral intersection'. *Computational Geometry & Applications,*Vol 4 No1 (March 1994).

[5] Zhu, X., Fang, S., and Bruderlin, B. 'Obtaining robust Boolean set operations for manifold solids by avoiding and eliminating redundancy'. *Proc. of 1993 ACM/IEEE 2nd Symposium on Solid Modelling and Applications.*

[6] Fang, S., Bruderlin, B., and Zhu, X. 'Robustness in solid modelling - a tolerance-based, intuitionistic approach'. *Computer-Aided Design*, Vol 25 No 9 (September 1993).

[7] Segal, M. 'Using tolerances to guarantee valid polyhedral modelling results'. *Computer Graphics* 24, 4 (1990).

[8] IGES (1991). 'The Initial Graphics Exchange Specification (IGES) version 5.1'. Iges/Pdes Organisation, National Institute of Standards and Technology, Gaithersburg, MD 20899, USA.

[9] STEP 'Standard for the Exchange of Product Model Data'. ISO 10303.