Boundary Representation Model Rectification

G. Shen, T. Sakkalis and N. M. Patrikalakis

Design Laboratory

Department of Ocean Engineering

Massachusetts Institute of Technology

Cambridge, MA 02139–4307, USA

Design Laboratory Memorandum 99-5

Copyright ©2001 Massachusetts Institute of Technology

All rights reserved

First submitted: November 18, 1999

Revised: March 5, 2001 and June 15, 2001

Abstract

Defects in boundary representation models often lead to system errors in modeling software and associated applications. This paper analyzes the model rectification problem of manifold boundary models, and argues that a rectify-by-reconstruction approach is needed in order to reach the global optimal solution. The restricted face boundary reconstruction problem is shown to be NP-hard. Based on this, the solid boundary reconstruction problem is also shown to be NP-hard.

Keywords: *Boundary reconstruction, NP-hardness,*

CAD model defects, robustness, data exchange.

1 Introduction

Model representations of products enable man-machine and machine-machine communication. With rules set by representation schemes, models can be realized as physical artifacts. However, such realization often fails whenever models contradict rules and ambiguities arise. This is especially true for manifold boundary representation (B-rep) as its validity is not self-guaranteed [1, 2, 3, 4]. In an earlier paper, we have identified sufficient conditions for validity of manifold B-rep models [5]. Defects in B-rep models are those representational features that do not conform to constraints set by modeling schemes. Such defects are topological and geometric errors, and visually appear as gaps, dangling faces, internal walls and inconsistent orientations. Defects may cause failures of modeling systems and applications because operations are typically designed with the assumption of model validity. Model rectification, a process that repairs defects, is essential to the success of design and manufacturing defect-free products in an integrated CAD/CAM environment.

Research on model rectification has been done mainly on triangulated models, specifically, STL models for rapid prototyping. STL models represent solids using oriented triangles [6]. Defects in STL models are gaps due to missing triangles, inconsistently oriented triangles, inappropriate intersections in the interiors of triangles. Most algorithms [7, 8] identify erroneous triangle edges, string such edges to form hole boundaries, and then fill holes with triangles. As pointed out in [7], topological ambiguities are resolved by intuitive heuristics. These algorithms [7, 8] use local topology (incidence and adjacency) to rectify defects and are successful in the majority of candidate models, but may create undesirable global topological and geometric changes. See also [9] for a critique of these methods.

Barequet and Sharir [9] developed a global gap-closing algorithm for polyhedral models, using a partial curve matching technique. In their method, gap boundaries are discretized. Each match between any two parts of gap boundaries is given a score based on the closeness of their discrete points. They have shown that finding a consistent set of partial curve matches with maximum score, a subproblem of their repairing process, is NP-hard. Barequet and Kumar [10] developed a model repairing system using the algorithm in [9], but with a modified score which is the normalized gap area between two matched parts of gap boundaries. Visualization tools were also provided to enable the user to override unwanted modifications. The system was improved by Barequet et al [11] in terms of efficiency, and extended to models with regular arrangement of entire NURBS surface patches. However, this extension does not handle trimmed patches with intersection curve boundaries and general B-rep models involving non-regular arrangement of surface patches.

A different type of global algorithm is based on spatial subdivision. Murali and Funkhouser [12] developed an algorithm which handles defects such as intersecting and overlapping polygons and mis-oriented polygons. The algorithm follows three steps: spatial subdivision, solid region identification, and model output. It first subdivides $\mathbf{R}^3$ into convex cells using planes on which polygons sit. A cell adjacency graph is then constructed. Each node is a convex cell, and each arc is a link between two cells sharing a polygon. Whether a cell is a part of the intended solid is determined by its solidity value, ranging from -1 to 1. The solidity value of a cell is computed based on how much area of the cell boundary is covered by original polygons as well as solidity values of neighboring cells. Boundary polygons of cells with positive solidity values are then output as the resulting solid boundary. A major advantage of this algorithm is that it always outputs a valid solid. One limitation is that it may mishandle missing polygons and add cells which do not belong to the model.

Hamann and Jean [13] proposed a user-assisted gap-closing method for curved boundaries using bivariate scattered data approximation techniques to approximate missing data in gap areas.

Another approach currently investigated by many researchers is the development of new geometric representations which are free of defects caused by the precision limitation of the computer [3, 14]. Two typical examples are *precise representation* and *error-bounded representation* (see [15, 16, 17]). The common characteristic of these two representations is that both use new arithmetic systems for computer-representation of numbers and the algebraic operations necessary for modeling are closed.

B-rep models contain topological and geometric specification of boundaries. Although rarely, topological errors could happen. Often a model has a geometric specification inconsistent with its topological specification. In such cases, it is unclear which one gives correct information about the boundary. *The task of model rectification, is to create a valid boundary model which is also intended by the designer.* Sakkalis et al [5] derived sufficient conditions for representational validity of ideal models. Such conditions are useful in developing defect identification and rectification methods. Krause et al [18] proposed a methodology, which views representational validity as an application-dependent concept, for processing (verifying and repairing) CAD data, especially those received from data exchange, and developed an experimental data processor for grid generation for aerodynamic simulation. Jackson [19] developed the concept of tolerant modeling implemented in the Parasolid modeller, where each face, edge and vertex of a B-rep model is associated with a local tolerance and these tolerances are taken into account in subsequent operations. Commercial software such as [20, 21] now repairs erroneous B-rep models interactively and/or semi-automatically. Such repairing tools usually succeed in fixing local defects but leave global consistency to the users or process in an iterative manner. This paper analyzes the nature of the manifold B-rep model rectification problem, as well as the complexity of the problem. It approaches the problem as a reconstruction problem – reconstruct a valid boundary model, which is also most likely to be the intended one, using only the information in the erroneous model.

The paper is organized as follows: Section 2 discusses the nature of the rectification problem, and argues that the problem should be approached as a *reconstruction problem* in order to reach the global optimal solution. Section 3 first studies and formulates a lower-dimensional problem, the face reconstruction problem, and proves that it is NP-hard. Then it extends this result to the boundary reconstruction problem. Section 4 concludes the paper. The paper also includes an appendix which provides a brief review of NP-completeness, needed in explaining this work.

2 Problem Statement

Defects must be identified before being rectified. Based on this principle, two approaches, local and global, may be explored for the development of rectification methods.

A *local rectification method* traverses through a representation of a model, identifies defects using the sufficient conditions derived in Sakkalis et al [5], and rectifies defects using the same sufficient conditions, or some heuristic rules and/or user assistance if ambiguities arise. Such an approach is the same as other rectification methods [7, 8, 20] except that conditions for identifying and rectifying defects may have some differences. However, a local method does not guarantee a global optimal solution. In addition, rectification of local defects may have unpredictable cascading effects on other topological and geometric entities in the model, and some defects may not be rectified without searching further. A *global rectification method*, on the other hand, identifies all defects once and for all. It then rectifies defects such that the resulting model is not only valid but also optimal based on some user-defined criteria. The main obstacle for developing such a method lies in the fact that a priori identification of *all* defects, done independently of rectification, is impossible in general. As the definition of a boundary is bottom-up in a model representation, the validity verification should proceed in the same manner. Higher dimensional entities cannot be verified if lower dimensional entities have not been rectified. Part of the reason why it is so difficult to implement an identify-and-rectify approach is the necessity to maintain the consistency between topological and geometric information. It is not difficult to identify a defect, but it is hard to decide what is right or wrong, whenever there is inconsistency between topological and geometric information.

The ultimate goal of model rectification is to find the model intended by the designer but misrepresented. However, without user assistance, any solution resulting from the erroneous model is only an educated guess. It comes down to the question "What do we trust about a model that contains defects?" This is the most fundamental question we must answer before we can move any further. For example, if a B-rep model has correct topological information, while there is no appropriate geometry embedded in the underlying surfaces, what is the intended boundary? Should the topological information be modified to accommodate the geometry, or should the geometry (surfaces) be perturbed to have a boundary consistent with the topological information?

To answer such questions, we need to classify the information in a boundary representation: those initialized or selected by the designer and those induced by the system. In a solid modeling system, the only entities the designer can directly manipulate are the underlying surfaces. All others, topological information and individual topological and geometric entities, are computed by the system. For this reason, we opt to believe that surfaces are the basic information to be used in a rectification method. Another reason for such a hypothesis is that surfaces are specifically designed to fulfill certain performance requirements and functionalities, and therefore, should not be subjected to any modification without the designer's permission. In addition to the above, another piece of information which can be used in the rectification process is the genus of the intended solid boundary. Genus captures the designer's intent and can be computed using the well-known Euler's formula [22].

In this context, model rectification becomes a model reconstruction problem. An algorithm for model rectification searches for the intended boundary in the union of all the surfaces, and rebuilds all necessary topological and geometric entities. However, there may exist many potential solid boundaries, or none at all, resulting from the surfaces. Without additional information, it would be difficult to make a choice. Since an erroneous model in a neutral format (e.g. STEP[23]) is computed with reasonable precision in its native system, the information in the model could help clarify such ambiguities, although it should be used with caution. Roughly speaking, a desirable solution is a model which describes a boundary somewhere "near" the object described by the original model, both topologically and geometrically.

A typical data structure for B-rep models consists of a topological structure and a geometric representation. For a simplified version, see Figure 1. The topological structure, shaded in Figure 1, is a graph which describes adjacency and incidence relations[1] (represented by arcs) between topological entities (represented by nodes). In typical implementations more detailed topological relations are explicitly stored than those implied in Figure 1 (e.g. adjacency relations between a face and all its neighboring faces). The geometric representation includes points, curve and surface equations, which are associated with appropriate topological entities. A model, thus, is an *instance* of the data structure, and is *valid* if it describes a solid boundary. In addition, a face is valid if it describes a set homeomorphic to a closed disk minus k mutually disjoint open disks, and has no handles. Also, see [5] for validity of other topological nodes.

Let m_o be a model with topological structure $G(m_o)$[2]. $G(m_o)$ is valid if it is possible to assign each topological entity (face, edge, vertex) a set of the corresponding dimension (surface, curve, point) whose interior is manifold, such that the union of these manifolds

[1] Two topological entities of different dimensionalities have an incidence relation if one is a proper subset of the other. Two topological entities of same dimensionality have an adjacency relation if their intersection is a lower dimensional entity that has an incidence relation with each of them.

[2] We denote models and face nodes by lowercase letters, and the point-sets they represent by uppercase letters. For example, model m_o represents solid M_o. The topological structure of a node is denoted by $G(node)$.

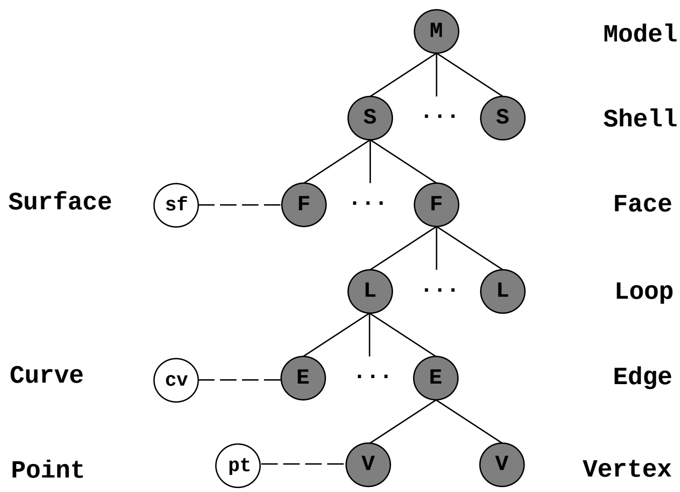

Figure 1: Data structure for B-rep models

bounds a solid. That is, if $G(m_o)$ is valid, there exists a nonempty set

$$\mathcal{M} = \{m \mid m \text{ is a valid model and has topological structure } G(m_o)\}. \tag{1}$$

For simplicity, in the following analysis, we assume that *models have only one shell.* It will be clear at the end of this paper that the same result applies to models with multiple shells. With this assumption, for any $m_1, m_2 \in \mathcal{M}$, M_1 is homeomorphic to M_2. Therefore, in case that the geometric representation of m_o is inconsistent with $G(m_o)$, if a reconstructed model m_n has topological structure $G(m_o)$, m_n is topologically equivalent to the model incorporating the design intent. If $G(m_n)$ is different from $G(m_o)$, the topological equivalence between m_n and m_o can be imposed by requiring that the genus of ∂M_n is equal to that of ∂M, where model $m \in \mathcal{M}$. We simply denote this by $g(m_n) = g(m_o)$, because both genuses can be computed by applying Euler's formula to $G(m_n)$ and $G(m_o)$, respectively.

Geometrically, two objects are close to each other if each one is in a neighborhood of the other. Some form of distance function could be used as a measure for this purpose, either the maximum distance or a well-defined average distance. An alternative, arguably more suitable for boundary rectification, is the boundary area change before and after rectification, because both the rectified and the original models use the same set of underlying surfaces. A correspondence can be established between a rectified face and an old face if they both have the same underlying surface, and the area difference between them measures the geometric change. No matter what measure is used, it should approach zero as the erroneous model becomes the exact model.

Since M_o and ∂M_o are not defined for a non valid model, we first define another set, called $\partial M_o'$ as follows. We project the loops of edges e_i of m_o onto each of the corresponding surfaces to obtain new loops of edges e_i' which bound new faces F_i', and let $\partial M_o' = \cup_I F_i'$. Note that $\partial M_o'$ may not bound a solid. The details of this construction are in Sections 3.1.1, 3.1.2 and 3.2. In these sections we define a function ϕ which evaluates the geometric difference between $\partial M_o'$ and ∂M_n. We will denote this by $\phi(\partial M_o', \partial M_n)$. Let ε be a user-specified tolerance for the geometric change and let m_o, $G(m_o)$ be as above. An ideal boundary reconstruction algorithm should follow the following procedure (see Figure 2):

1. Find a new model m_n, such that m_n has topological structure $G(m_o)$, i.e. $G(M_n) \sim G(m_o)$[3] and $\phi(\partial M_o', \partial M_n) \leq \varepsilon$.

[3] Two graphs are homeomorphic if both can be obtained from the same graph by a sequence of subdivisions of arcs.

2. If there exists a number of such new models, select the one with the minimal ϕ value.

3. Otherwise, find a new model m_n, such that $G(m_n)$ is different from $G(m_o)$ but $g(m_n) = g(m_o)$, and $\phi(\partial M'_o, \partial M_n) \leq \varepsilon$.

4. If there exists a number of such new models, select the one with the minimal topological structure change, e.g. the difference of the total numbers of arcs and nodes in $G(m_n)$ and $G(m_o)$ is minimal.

5. Otherwise, find a new model m_n with $\phi(\partial M'_o, \partial M_n) \leq \varepsilon$. If there exist more than one such new boundaries, select the one with the minimal topological change (i.e. minimal genus change); otherwise, no new model is reconstructed.

In the next section, we study the following subproblem which is essential to this reconstruction process:

Boundary reconstruction (BR) problem: Given a B-rep model m_o, whose geometric representation is inconsistent with its topological structure, reconstruct a new model m_n using only the information in m_o, such that: (1) $g(m_n) = g(m_o)$ and (2) $\phi(\partial M'_o, \partial M_n)$ is minimal.

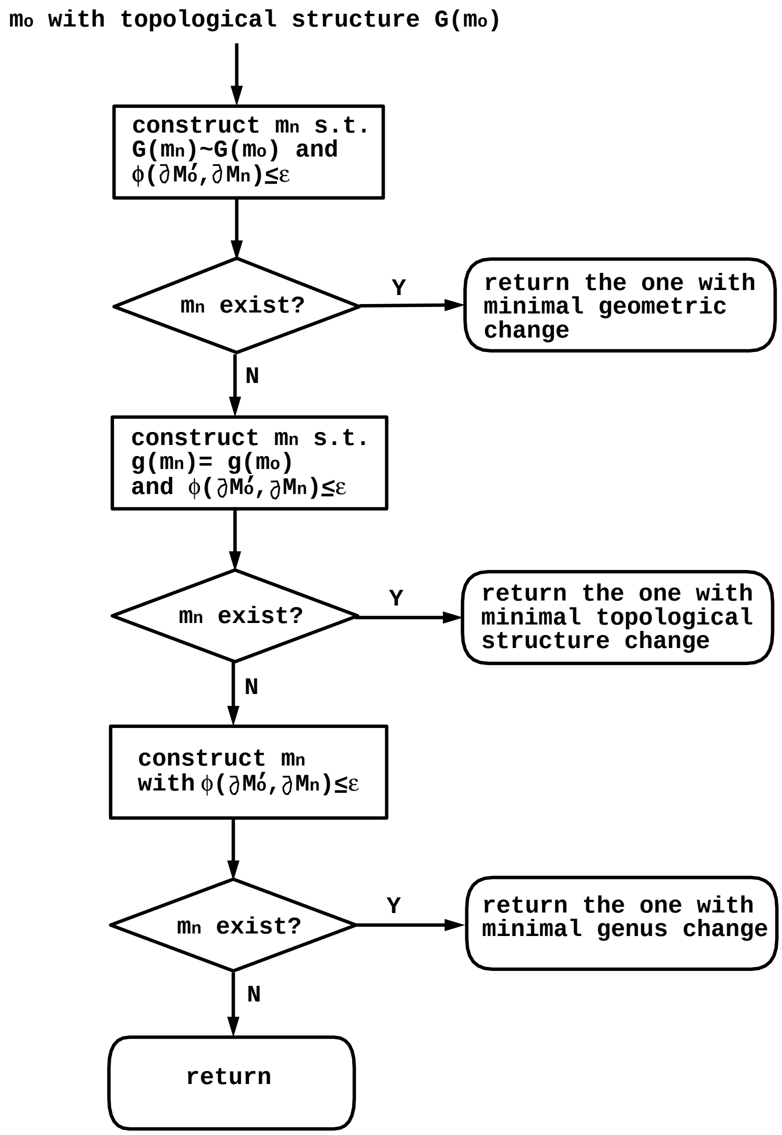

Figure 2: Flow chart of an ideal global reconstruction algorithm

3 Problem Complexity

3.1 Face reconstruction problem

Before moving onto the boundary reconstruction (BR) problem, we first study a lower-dimensional problem which not only gives us insight on the nature of such reconstruction problems, but also plays a crucial role in understanding the complexity of the BR problem.

A face node in a B-rep model represented in a certain format such as STEP [23] is likely to have inconsistent geometric features such as edges whose underlying curves are not on the underlying surface of the face. Topological errors such as open loops make a clear definition of the face geometry even more elusive. Similar to boundary reconstruction, face reconstruction builds a new face node f_n from an erroneous face node f_o, using only the information in the given model, such that f_n is not only valid but also close to the object described by f_o in both topology and geometry. We formulate the face reconstruction problem as an analog to the BR problem.

A valid face is homeomorphic to a closed disk minus k mutually disjoint open disks, and has no handles [5]. If f_o is valid, $G(f_o)$ is a planar graph that consists of simple cycles. Any two of these cycles may share at most one common nodes. Two graphs are homeomorphic if both can be obtained from the same graph by a sequence of subdivisions of arcs [24]. However, two homeomorphic graphs may have different geometric embeddings, and thus define two faces which are not homeomorphic. See Figure 3. In order to capture the design intent topologically, the component containing the outer loop in $G(f_n)$ also needs to be homeomorphic to that in $G(f_o)$, in addition to that $G(f_n)$ is homeomorphic to $G(f_o)$, so that the above situation is prevented. In this case we say that $G(f_n)$ is homeomorphic to $G(f_o)$ in the *strong sense*. In the following problem statement, ϕ_f is a function evaluating the geometric change before and after rectification, and will be elaborated in Section 3.1.2.

Face reconstruction (FR) problem: Given a face node f_o, whose geometric representation is inconsistent with its topological structure, in a B-rep model m_o, reconstruct a valid face node f_n, using only the information in m_o, such that $G(f_n)$ is homeomorphic to $G(f_o)$ in the strong sense and $\phi_f(F_n, F_o)$ is minimal.

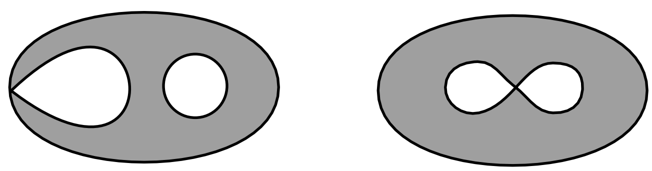

Figure 3: Two homeomorphic graphs with different geometric embeddings

In the following, we first study how an invalid face node could be rectified, and then formulate the face reconstruction problem mathematically and prove its NP-hardness.

3.1.1 Face boundary reconstruction

Let R be the underlying surface of face node f_o, and $\{R_i\}_{1\leq i\leq N}$ be surfaces in m_o such that $C_i = R \cap R_i \neq \emptyset$. Surface intersections could be very complex. Here, for simplicity, we assume that surfaces do not overlap. In addition, we also exclude isolated intersection points, since such a point does not bound a finite region on R, and therefore, is not used to form the face boundary. However, if an intersection point between two surfaces is on an intersection curve, it may be used as a vertex in the face boundary. See Figure 4(a), where the intersection point of R and R_4 is on C_1.

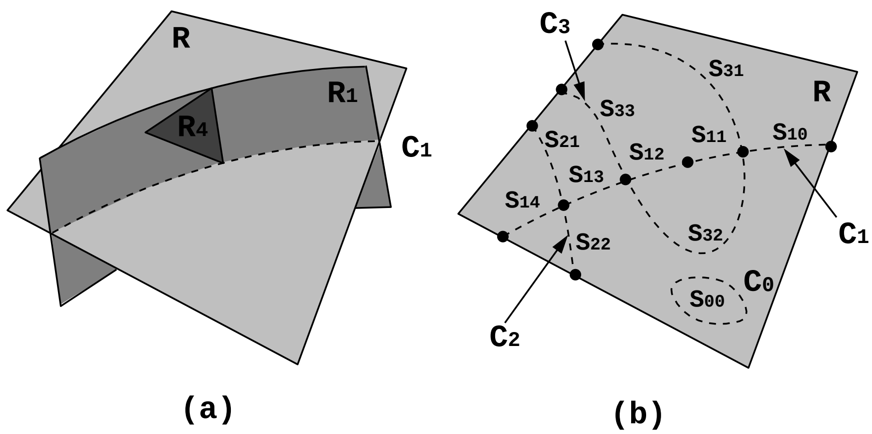

Figure 4: Curve segments from surface intersections

Furthermore, C_i's may intersect each other. Each C_i is subdivided by intersection points into curve segments, each of which is either an open curve bounded by two intersection points or a simple closed curve. Those intersection points, indeed, are intersections of three or more surfaces. Denote the collection of curve segments on all C_i's by $\{S_{ij}\}$. See Figure 4(b), where broken lines are curve segments and solid dots are intersection points. In the figure, because the underlying surface is finite, its boundary curves are also used in creating curve segments. Notice that $\{S_{ij}\}$ subdivide R into patches whose interiors are disjoint and intersection-free. The face geometry must be either one of these patches or the union of some patches. Therefore, the face boundary consists of curve segments from $\{S_{ij}\}$.

In the following description, for simplicity, we use a lowercase symbol to denote an edge or a vertex as a point-set. Whenever the corresponding representational node is referred, the word *node* is used before the notation. For example, node e represents edge e. For models and faces, the same notation scheme as in the previous section is used, i.e. lowercase symbols for nodes and uppercase symbols for point-sets.

Let node e_{i_k} be an edge node in f_o, also shared by face node f_i having R_i as its underlying surface. Then, to maintain geometric consistency of the adjacency relation between these two faces, e_{i_k} must be a subset of C_i. This is rarely true as the underlying curve given in node e_{i_k} is often an approximation of the exact intersection curve C_i. As a matter of fact, as illustrated in Figure 5, e_{i_k} may be pathologically defined by a space curve (the broken line) and two points (the two circles) which may not be on the curve as they are supposed to be. Because the adjacency relation is symbolic and thus exact, the initial rectification of node e_{i_k} can be done by using C_i as the underlying curve and discarding the one given in the original model. Consequently, the vertices must be on C_i. They also need to be close to their original erroneous positions in order to reflect the design intent. Reasonable replacements of the original vertices v_{i1}, v_{i2}, for instance, could be the projections v'_{i1}, v'_{i2} of v_{i1}, v_{i2} onto C_i. See Figure 5, where the new vertices are solid dots. Therefore, such a rectified edge, denoted by e'_{i_k}, is a subset of C_i, bounded by v'_{i1}, v'_{i2} and oriented in the same way as e_{i_k} provided that the given underlying curve in node e_{i_k} and C_i are not far apart.

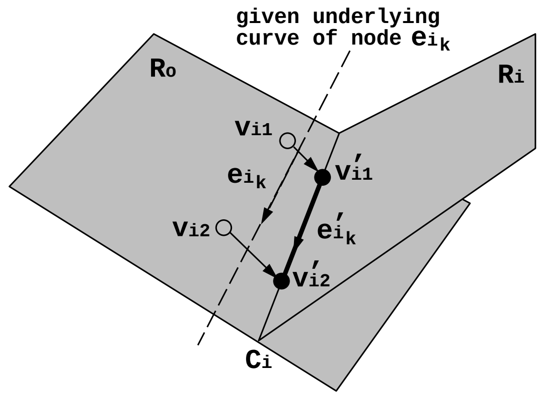

Figure 5: Initial rectification of an edge on a face

However, as a face should be bounded by curve segments from $\{S_{ij}\}$, a rectified edge should consist of such curve segments. Edge e'_{i_k} is not guaranteed to be so. See Figure 6(a). Further rectification of e_{i_k} selects some curve segments on C_i such that their union is a simple open or closed curve and an optimal approximation of e'_{i_k}. The union defines a new edge e''_{i_k}, whose corresponding node has the same symbolic information as node e_{i_k}, i.e. the same embedding surfaces, the same parent faces, and the same orientation, but has a consistent geometric representation while node e_{i_k} does not. Edge e''_{i_k} can be obtained by perturbing vertices v'_{i1}, v'_{i2} of e'_{i_k} to the closest intersection points on C_i. This vertex perturbation is also necessary in order to achieve geometric consistency at a vertex. A vertex is involved in various incidence and adjacency relations between its incident faces and edges, and therefore, needs to be positioned at the intersection point of those underlying surfaces.

The collection of such rectified edges, however, may not form a valid face boundary, as there

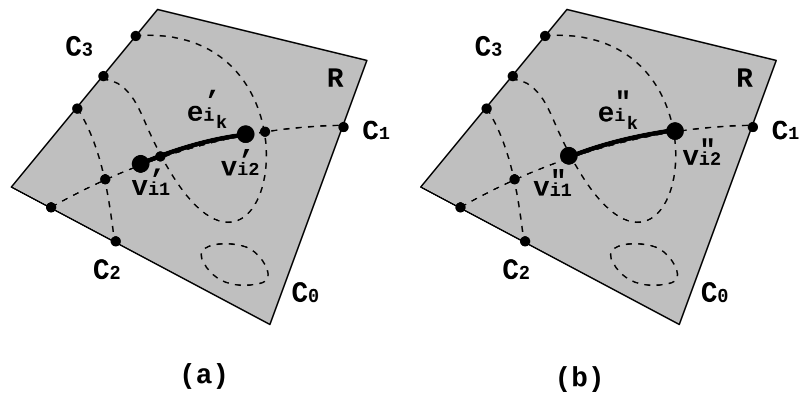

Figure 6: Creation of e''_{i_k} by vertex perturbation

may exist open loops and/or dangling curve segments. For example, in Figure 7(a), (where the rectified edges are thick lines,) a dangling curve segment S_{12} and a gap between e''_5 and e''_9 exist. The final act of face reconstruction, is to trim away dangling curve segments and fill gaps with curve segments from $\{S_{ij}\}$, so that the selected curve segments form a valid face boundary. See Figure 7(b). This trimming and gap-filling process should create a face boundary which satisfies the topological and geometric requirements stated in the problem statement.

Assume that in the final reconstructed face boundary, the selected curve segments from C_i are

$\mathcal{S}_i = \{S_{ij_l}\}_{1\le l\le L_i}$. A new edge $e^n_{i_k}$ is created by stringing curve segments in $\mathcal{S}_i$. The process starts with an arbitrary curve segment $S_{ij_1} \in \mathcal{S}_i$, and searches for its adjacent curve segments in $\mathcal{S}_i$. If, at one endpoint, exactly one adjacent curve segment S_{ij_2} is found, then, S_{ij_2} is selected and the search marches on at the other endpoint of S_{ij_2}; otherwise,

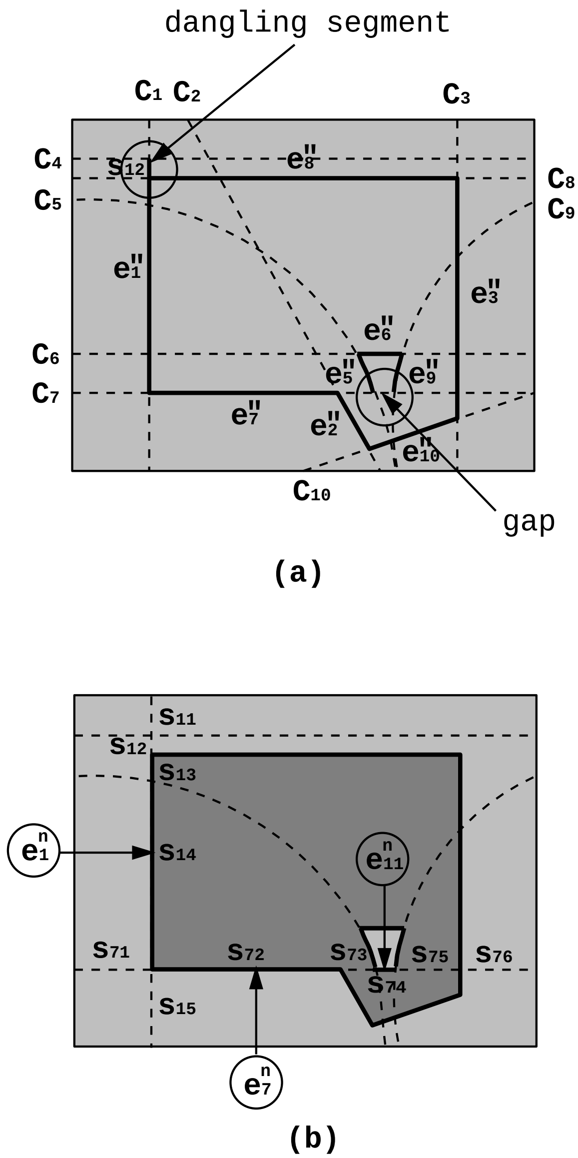

Figure 7: Trimming of dangling curve segments (a) and gap filling (b)

the process terminates and resumes at the other endpoint of S_{ij_1}. Notice that there may be more than one new edges created from $\mathcal{S}_i$. For example, in Figure 7, curve segments S_{72}, S_{74} on C_7 are selected in the final face boundary, creating two new edges e_7^n and e_{11}^n.

In summary, an edge e_{i_k} in the original model, shared by faces f_o and f_i, is first rectified by projecting its vertices onto $C_i = R \cap R_i$, which is the exact underlying curve of node e_{i_k}. This produces e'_{i_k}, and achieves a consistent adjacency relation between f_o and f_i but may not do so at the vertices. Edge e'_{i_k} is then rectified by perturbing its vertices to their closest intersection points on C_i, such that the resulting edge e''_{i_k} consists of curve segments in $\{S_{ij}\}$. Notice that node e''_{i_k} has the same symbolic information as node e_{i_k} and could be geometrically consistent with all adjacency and incidence relations in which it is involved, if other topological entities are appropriately constructed. Typically, the geometric change between e''_{i_k} and e_{i_k} is minimal. Since such edges may not form a valid face boundary, additional curve segments from $\{S_{ij}\}$ are added to fill gaps and dangling curve segments are trimmed away. In the rectified face boundary, a new edge $e^n_{i_k}$ consists of curve segments from C_i.

3.1.2 Problem formulation and proof of NP-hardness

To mathematically formulate the FR problem, especially to quantify face geometric change, we classify new edges into two categories:

1. A new edge $e^n_{i_k}$ belongs to the first category if it has a corresponding edge e_{i_k} in the

original model, i.e. $e^n_{i_k}$ is considered to be the rectified e_{i_k}. Such a new edge node has the same symbolic information as node e_{i_k} and a geometry which is an optimal approximation of that of node e_{i_k}. Because e_{i_k} is not well defined, the geometric change between $e^n_{i_k}$ and e_{i_k} is measured by comparing $e^n_{i_k}$ and e'_{i_k}, which is not only well defined but also symbolically the same as and geometrically close to e_{i_k}. We define $\phi_e : (C_i \times C_i) \rightarrow \mathbf{R}$, where

$$\begin{aligned} \phi_e(e^n_{i_k}, e'_{i_k}) &= length((e^n_{i_k} \cup e'_{i_k}) - (e^n_{i_k} \cap e'_{i_k})), \\ & e^n_{i_k}, e'_{i_k} \subseteq C_i. \end{aligned} \tag{2}$$

For example, in Figure 6, if $e^n_{i_k} = e''_{i_k}$, meaning that during the trimming-and-filling stage the edge is unchanged, then

$$\phi_e(e^n_{i_k}, e'_{i_k}) = |v'_{i1}v''_{i1}| + |v'_{i2}v''_{i2}|, \tag{3}$$

where $|\ |$ denotes the length of a curve segment. If there exist more than one new edges on C_i which could belong to the first category, the one with the minimal ϕ_e value is designated as the new edge in the first category, i.e. there is at most one corresponding new edge in the new boundary for each edge in the original model. In addition, $e^n_{i_k} \cap e'_{i_k} \neq \emptyset$, so that a new edge whose geometry is far away from e'_{i_k} will not be taken as the corresponding new edge of e_{i_k}. This could happen, for example, when two new edges $e^n_{i_k,1}, e^n_{i_k,2}$ have the same symbolic information as e_{i_k} but

$$e^n_{i_k,1} \cap e'_{i_k} \neq \emptyset,$$

$$e^n_{i_k,2} \cap e'_{i_k} = \emptyset,$$

$$\phi_e(e^n_{i_k,1}, e'_{i_k}) > \phi_e(e^n_{i_k,2}, e'_{i_k}).$$

In Figure 7(b), all the new edges, except e^n_{11}, belong to the first category.

2. All other new edges belong to the second category. These edges are generally added to close gaps. Let a new edge be on curve $C_j = R \cap R_j$. It is possible that the face on R_j is not adjacent to f_o in the original model, and therefore, the new edge on C_j does not have a corresponding old edge.

Because the edges in the original face node carry the design intent, geometric changes to them should be minimized, i.e. $\sum \phi_e$ for the new edges of the first category should be minimal. If there exist more than one choices of new face boundary having the minimum, the one with the shortest total length of the edges of the second category should be chosen, because any drastic change is not trustworthy. Figure 8(a) shows the curve segments and initially rectified edges. Notice that the original face node has an inconsistent geometric representation. It can be seen in Figure 8(b) that if all four edges represented by thick lines are selected in the final boundary, $\sum \phi_e$ is the minimum. There exist five such loops. Figure 8(c) shows the final face boundary which has the shortest gap-closing edge e^n_5, and Figure 8(d) shows the other four.

We now formulate the face reconstruction problem as a search problem:

Face reconstruction (FR) problem: Let f_o be a face node, whose geometric representation is inconsistent with its topological structure, in a B-rep model, and $\{S_{ij}\}$ be as

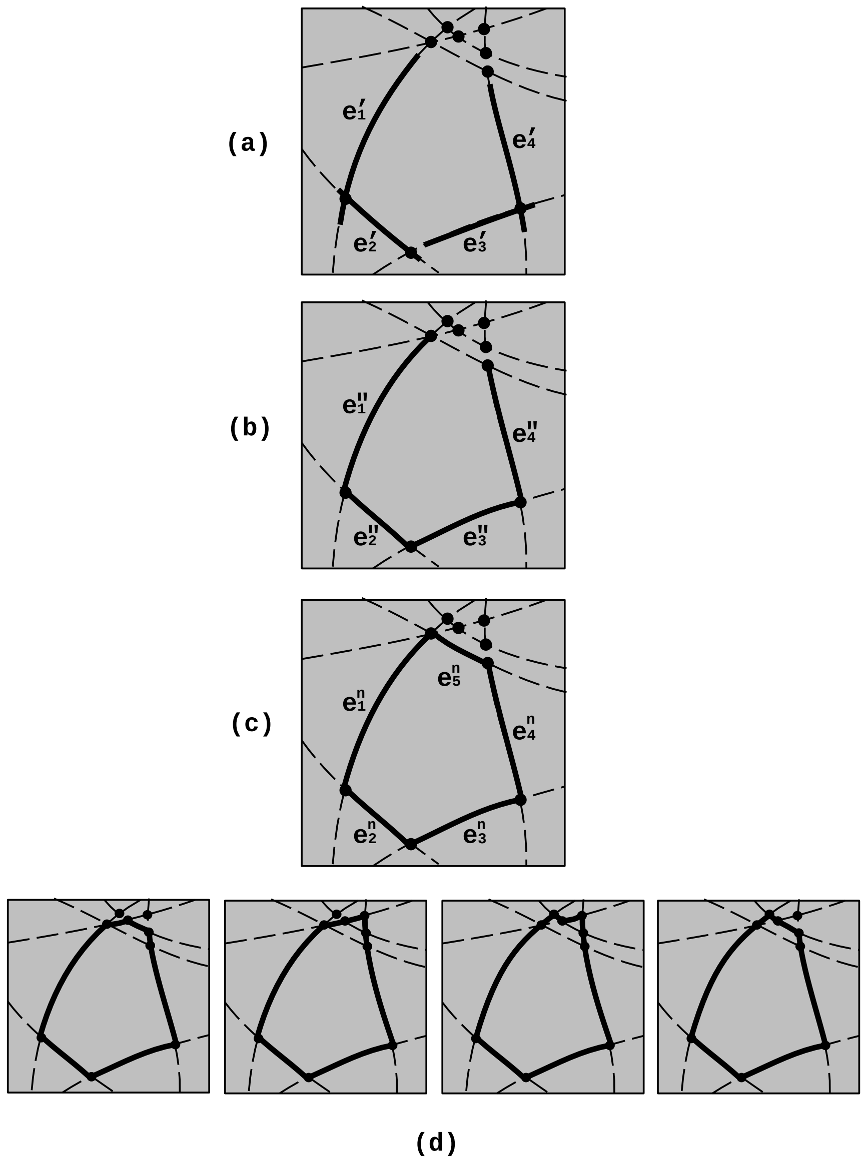

Figure 8: Minimization of edge geometric change

above. Search for a subcollection

$$\{S_{ij_l}\}_{1\leq i\leq N, 1\leq l\leq L_i} \tag{4}$$

of $\{S_{ij}\}$, where L_i is the number of curve segments selected on C_i, such that

1. $\{S_{ij_l}\}$ bounds a face F_n.
2. $G(f_n)$ is homeomorphic to $G(f_o)$ in the strong sense.
3. $\sum \phi_e(e'_{i_k}, e^n_{i_k})$ is minimal for the new edges of the first category.
4. If condition (3) is satisfied, then, the total length of the edges of the second category is also minimal.

Intuitively, this problem can be converted to a graph problem as $\{S_{ij}\}$ forms a geometric embedding on R of a graph. This graph, $G_f = (V_f, E_f)$, has curve segments in $\{S_{ij}\}$ as arcs in E_f and intersection points as nodes in V_f. The solution to the FR problem is then a subgraph satisfying the properties in the problem statement. The face reconstruction process is a process of searching for such a subgraph. A straightforward implementation of this process is to search all possible subgraphs and find the one satisfying conditions 1 to 4 above. Such an exhaustive search, however, may need exponential time in terms of the number of arcs in E_f. In the following, we prove that the FR problem is NP-hard by proving a restricted problem is NP-hard. See also the appendix for some further comments on NP-hard problems. But first, we introduce a known NP-hard problem [25]:

Rural Postman Problem:[4] Let graph $G = (V, E)$. Each $e \in E$ has length $l(e) \in Z_0^+$. Let $E' \subseteq E$. Find the circuit in G that includes each arc in E' and that has the shortest total length.

Theorem 3.1 *FR problem is NP-hard.*

Proof: The basic idea of the proof is to consider the following instance of the restricted FR problem: The topological structure of f_o represents a closed disk, and there is only one curve segment on each C_i (see Figure 9). The boundary of F_o is then a simple closed curve. The solution to the problem must be a circuit in graph G_f if condition (2) is to be satisfied.

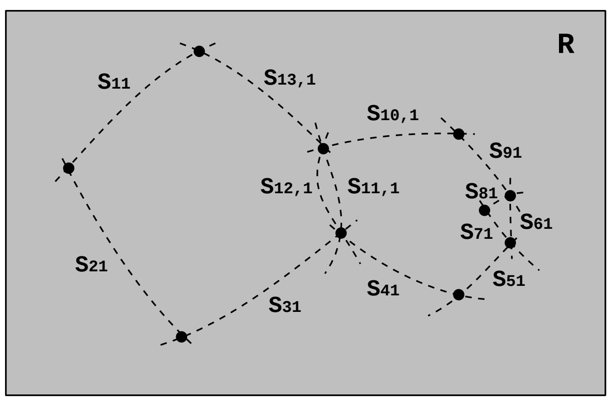

Figure 9: An instance of the restricted FR problem

For each edge e_{i_k} of F_o, we construct e'_{i_k}, e''_{i_k} as above. Because there is only one curve segment S_{i1} on each C_i, $e''_{i_k} = S_{i1}$ provided that e'_{i_k} is not far different from S_{i1}. If S_{i1}

[4]Here we give the Rural Postman search problem because the FR problem is also a search problem. In [25], the Rural Postman decision problem is given.

is in the reconstructed face boundary, then $e^n_{i_k} = e''_{i_k} = S_{i1}$, and therefore, $\phi_e(e^n_{i_k}, e'_{i_k})$ is minimal, due to the way e''_{i_k} is constructed. If it is not in the face boundary, then $\phi_e(e^n_{i_k}, e'_{i_k})$ is maximal, because $e^n_{i_k} = \emptyset$. Let

$$\mathcal{S}_1 = \{S_{i1} | \text{ there exists } e''_{i_k} = S_{i1}, 0 \le i \le N.\} \quad (5)$$

Then, if all the curve segments in $\mathcal{S}_1$ are selected, $\sum \phi_e$ for the edges of the first category reaches its minimum. This means that a valid face boundary which includes all the curve segments in $\mathcal{S}_1$, will be selected over any choice which does not.

The corresponding instance of the Rural Postman problem is as follows: Graph G_f with $l(e) = length(S_{i1})$ for each i, and $E' = \mathcal{S}_1$. We prove that it can be reduced to the restricted FR problem at least in the abstract setting of graph theory [5].

It can be observed that the solution to the restricted FR problem answers the Rural Postman problem; if the solution exists for the restricted FR problem and contains all the curve segments in $\mathcal{S}_1$, it is the circuit with the shortest length and including all the arcs in E', and therefore, the solution to the Rural Postman problem; if the solution does not exist or it exists but does not contain all the edges in $\mathcal{S}_1$, no solution exists for the Rural Postman problem. ■

[5]Theoretically, it is possible to develop a linear algorithm to draw a planar graph on the plane. See [26, 27]. This establishes the argument that an abstract graph search problem could be converted to an instance of the geometric problem (FR problem).

3.2 Boundary reconstruction problem

Now we formulate the BR problem and prove that it is also NP-hard.

Let m_o be the given B-rep model. The underlying surfaces, $\{R_i\}_{1 \leq i \leq N}$, are subdivided by surface intersections into a collection of patches

$$\{P_{ij}\}_{1 \leq i \leq N, 1 \leq j \leq N_i}, \tag{6}$$

where N_i is the number of patches on surface R_i. For a face node f_i, F_i may not be well defined. Because the embedding information is symbolic and thus exact, the initial rectification of F_i can be done by trimming the underlying surface R_i of f_i using projections of the loops in f_i onto R_i. Denote such a face by F_i'. As in face reconstruction, F_i' is further rectified by selecting patches from $\{P_{ij}\}_{1 \leq j \leq N_i}$ to form a new geometry which is an optimal approximation of F_i'. Denote this new face by F_i''. The difference between F_i' and F_i'' can be measured by function $\phi_f : (R_i \times R_i) \rightarrow \mathbf{R}$, where

$$\begin{aligned} \phi_f(F_i', F_i'') &= area((F_i' \cup F_i'') - (F_i' \cap F_i'')), \\ & F_i', F_i'' \subseteq R_i. \end{aligned} \tag{7}$$

Such rectified faces may not form a valid solid boundary due to the possible existence of dangling patches and holes. Therefore, a trimming-and-filling process follows. Similar to face reconstruction, in the new solid boundary, a new face F_i^n belongs to the first category if it has a corresponding old face, and to the second category if it does not. For the faces of the first category, $\sum \phi_f(F_i^n, F_i')$ should be minimized. If there exist more than one valid boundaries having the minimum $\sum \phi_f$, that with the minimal total area of the faces of the second category should be chosen.

The boundary reconstruction problem can also be formulated as a search problem:

Boundary Reconstruction (BR) problem: Let m_o be a B-rep model whose geometric representation is inconsistent with its topological structure, and $\{P_{ij}\}$ be as above. Search for a subcollection

$$\{P_{ij_k}\}_{1\le i\le N, 1\le k\le K_i}, \tag{8}$$

of $\{P_{ij}\}$, where K_i is the number of patches selected on surface R_i, such that

1. $\{P_{ij_k}\}$ bounds a solid M_n.
2. $g(m_n) = g(m_o)$, where m_n is the new model representing M_n.
3. $\sum \phi_f(F_i^n, F_i')$ is minimal for the faces of the first category.
4. If condition (3) is satisfied, then, the total area of the faces of the second category is also minimal.

We now prove that the BR problem is NP-hard:

Theorem 3.2 *BR problem is NP-hard.*

Proof: We prove the theorem by converting the restricted FR problem to the BR problem. Assume that in an instance of the restricted FR problem, the face node f_o has a plane as its underlying surface. Sweep the face along the normal direction to a parallel plane. The sweeping solid of F_o should be homeomorphic to a closed ball. The instance of the BR problem is a patch collection

$$\begin{aligned} \{P_{ij}\} &= \{\text{patches generated from curve segments}\} \cup \\ &\quad \{\text{patches from the two planes}\}, \end{aligned} \tag{9}$$

and a model m_o with its topological structure representing a sphere and $\{F_i'\}$ generated from $\{e_{i_k}'\}$. See Figure 10 for an example. This conversion can be executed in polynomial time. If a subcollection of $\{S_{ij}\}$ is the solution to the restricted FR problem, then the patches generated from the curve segments in the subcollection, with additional patches from the two planes, is the solution to the BR problem. Conversely, if a subcollection of the patches is the solution to the BR problem, it must be bounded by two patches from the two planes and patches whose generating curve segments indeed form the solution to the restricted FR problem. ■

For models with multiple shells, the same result holds, because the boundary reconstruction problem of models with one shell is a special case of that of models with multiple shells.

4 Conclusions

The BR problem is, in certain ways, similar to the gap boundary matching problem studied by Barequet and Sharir [9] which is also shown to be a NP-hard (global search) problem. However, they are quite different in nature as the former reconstructs a boundary from

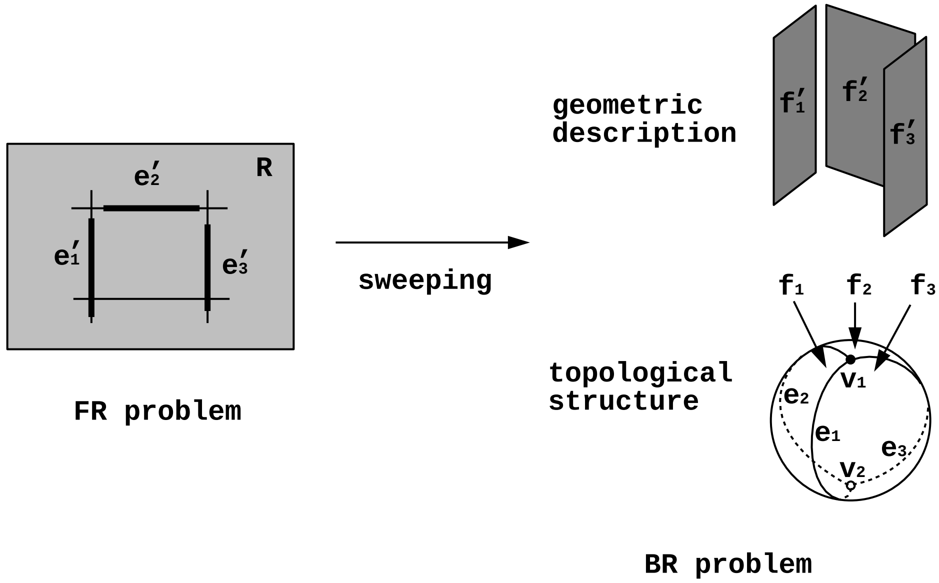

Figure 10: Illustration of the proof of Theorem 3.2

a set of surfaces and involves complex topological constraints, whereas the latter matches boundaries of a set of surface patches. Theoretically, of course, all NP-complete problems are equivalent, but their specifics can vary significantly.

The understanding of the NP-hardness of a problem helps design algorithms to efficiently and properly solve the problem. In a recent paper [29], we propose model rectification algorithms which approximate the process of finding the optimal solutions described here. To achieve numerical robustness in floating point environment, interval model representation proves to be a powerful tool [16, 17]. We further study some key topological issues of interval solid models in [28], and develop interval geometric and numerical methods useful in model rectification in [30]. All these are summarized in a review paper [31].

Appendix: Short review of NP-completeness

This appendix provides a brief introduction to the theory of NP-completeness. For details, see [25]. A decision problem is a problem whose answer is *yes* or *no*. A problem is said to belong to the class P if it can be solved by a polynomial time DTM program. DTM is the abbreviation for *deterministic one-tape Turing machine*, a simplified computing model. If the problem can be solved by a polynomial time NDTM program, it belongs to the class NP. NDTM, *non-deterministic one-tape Turing machine*, has the exact same structure as DTM, except that it has a *guessing module*. A NDTM program has two distinct stages: guessing and checking. For example, the traveling salesman problem is given as:

Traveling salesman problem: Given a finite set of cities, a distance between each pair of cities, and a bound B, is there a tour of all the cities such that the total distance of the tour is no larger than B?

A NDTM program first guesses a tour of all the cities, and then verifies in polynomial time if the length of this guessed tour is less than the given threshold. If at least one guessed tour is accepted, the answer to the problem is *yes*. Therefore, the traveling salesman problem is in class NP.

It is observed that class(P)$\subseteq$class(NP). It is also an unproven conjecture that P$\neq$NP. For a problem in NP, there exists a polynomial p such that the problem can be solved by a deterministic algorithm having time complexity $O(2^{p(n)})$, where n is the length of the input string. A decision problem Π is NP-complete if $\Pi \in$NP and, for all other decision problems

$\Pi' \in$NP, there is a polynomial transformation from Π' to Π. A polynomial transformation is a mapping f such that

1. There exists a polynomial DTM program which computes f.

2. For any instance x of the problem, x is accepted if and only if $f(x)$ is accepted.

Therefore, NP-complete problems are the "hardest" problems in NP. The NP-completeness proof of a decision problem consists of the following four steps:

1. Show that $\Pi \in$NP, i.e. Π can be solved by a NDTM program.

2. Select a known NP-complete problem Π'.

3. Construct a transformation f from Π' to Π.

4. Prove that f is a polynomial transformation.

The NP-completeness can also be proven by restriction, that is, a problem is NP-complete if it contains a known NP-complete problem as a special case.

The concept of NP-hardness applies to problems outside class NP, e.g. search problems. Informally, a search problem is NP-hard if it is at least as hard as some NP-complete problem. For example, the search of the shortest tour of all the cites is as hard as the traveling salesman decision problem, because the solution to the search problem certainly answers the decision problem.

Acknowledgments

Funding for this work was obtained in part from NSF grant DMI-9215411, ONR grant N00014-96-1-0857 and from the Kawasaki chair endowment at MIT. The authors would like to thank W. Cho and G. Yu for their assistance in the early phases of the CAD model rectification project. We also would like to thank T. J. Peters, A. C. Russell, the referees and the area editor of the Graphical Models journal for their comments on this work.

Note: This paper was also presented at the 6th ACM Symposium on Solid Modeling and Applications, Ann Arbor, MI, June 2001.

References

[1] A. A. G. Requicha, Representations for rigid solids: Theory, methods, and systems, *ACM Computing Surveys*, **12**(4), 1980, 437–464.

[2] D. E. LaCourse, *Handbook of Solid Modeling*, McGraw-Hill Inc., 1995.

[3] C. M. Hoffmann, *Geometric and Solid Modeling–An Introduction*, Morgan Kaufmann Publishers Inc., San Mateo, CA, 1989.

[4] M. Mäntylä, *An Introduction to Solid Modeling*, Computer Science Press, Rockville, Maryland, 1988.

[5] T. Sakkalis, G. Shen and N. M. Patrikalakis, Representational Validity of Boundary Representation Models. *Computer Aided Design*, **32**(12), October 2000, 719–726.

[6] 3D Systems, Inc., *Stereolithography Interface Specification*, June 1988.

[7] J. H. Bohn and M. J. Wozny, A topology-based approach for shell-closure, in *Geometric Modeling for Product Realization* (P. R. Wilson, M. J. Wozny and M. J. Pratt, Eds.), pp. 297–318, Elsevier Science Publishers BV, 1993.

[8] I. Makela and A. Dolenc, Some efficient procedures for correcting triangulated models, in *Proceedings of Solid Freeform Fabrication Symposium*, University of Texas at Austin, 1993, pp. 126–134.

[9] G. Barequet and M. Sharir, Filling gaps in the boundary of a polyhedron, *Computer Aided Geometric Design*, **12**(2), 1995, 207–229.

[10] G. Barequet and S. Kumar, Repairing CAD Models, in *Proc. IEEE Visualization*, Phoenix, Arizona, 1997, pp. 363–370.

[11] G. Barequet, C. A. Duncan, and S. Kumar, RSVP: A Geometric Toolkit for Controlled Repair of Solid Models, *IEEE Transactions on Visualization and Computer Graphics*, **4**(2), 1998, 162–177.

[12] T. M. Murali and T. A. Funkhouser, Consistent solid and boundary representations from arbitrary polygonal data, In A. Van Dam, editor, *Proceedings of 1997 Symposium*

on Interactive 3D Graphics, Providence, Rhode Island, 1997. New York, ACM Press, pp. 155-162.

[13] B. Hamann and B. A. Jean, Interactive surface correction based on a local approximation scheme, *Computer Aided Geometric Design*, **13**(4), 1996, 351–368.

[14] C. M. Hoffmann, The problems of accuracy and robustness in geometric computation, *Computer*, **22**(3), 1989, 31–41.

[15] S. Fortune, Polyhedral modeling with multiprecision integer arithmetic, *Computer Aided Design*, **29**(2), 1997, 123–133.

[16] C.-Y. Hu, N. M. Patrikalakis and X. Ye, Robust interval solid modeling: part I, representations, *Computer Aided Design*, **28**(10), 1996, 807–817.

[17] C.-Y. Hu, N. M. Patrikalakis and X. Ye, Robust interval solid modeling: part II, boundary evaluation, *Computer Aided Design*, **28**(10), 1996, 819–830.

[18] F.-L. Krause, C. Stiel and J. Lüddemann, Processing of CAD-data – conversion, verification and repair, in *Proceedings of 4th Symposium on Solid Modeling and Applications*, Atlanta, GA, May 1997, C. Hoffmann and W. Bronsvoort, eds., pp. 248–254. NY: ACM, 1997.

[19] D. J. Jackson, Boundary Representation Modelling with Local Tolerances, *Proceeding of the 3rd Symposium on Solid Modeling and Applications.* Salt Lake City, Utah, May 1995. C. Hoffmann and J. Rossignac, eds., pp. 247–253, NY: ACM, 1995.

[20] International TechneGroup Incorporated, *Introduction – CADfix Makes Geometry Data Exchange Work*. http://www.iti-oh.com/

[21] Theorem Solutions Inc, *Product Information*, http://www.theorem.co.uk/productmain.htm

[22] I. C. Braid, *Notes on a geometric modeller*, C.A.D. Group Document No. 101, University of Cambridge, UK, 1979.

[23] U. S. Product Data Association, *ANS US PRO/IPO-200-042-1994: Part 42 – Integrated Geometric Resources: Geometric and Topological Representation*, 1994.

[24] F. Harary, *Graph Theory*, Addison-Wesley Publishing Company, 1969.

[25] M. R. Garey and D. S. Johnson, *Computer and Intractability: A Guide to the Theory of NP- completeness*, W. H. Freeman and Company, 1979.

[26] N. L. Biggs and A. T. White, *Permutation Groups and Combinatorial Structures*, Cambridge University Press, 1979.

[27] N. Chiba, T. Yamanouchi and T. Nishizeki, Linear Algorithms for Convex Drawings of Planar Graphs, in *Progress in Graph Theory* (J. A. Bondy and U. S. R. Murty, eds.), pp. 153–173, Academic Press, 1984.

[28] T. Sakkalis, G. Shen and N. M. Patrikalakis, Topological and Geometric Properties of Interval Solid Models, *Graphical Models*, **63**(2), 2001.

[29] G. Shen, T. Sakkalis and N. M. Patrikalakis, Manifold Boundary Representation Model Rectification ('La rectification des modèles des varietés B-rep'), *Proceedings of the 3rd International Conference on Integrated Design and Manufacturing in Mechanical Engineering.* C. Mascle, C. Fortin, J. Pegna, editors. Page 199 and CDROM. Montreal, QC, Canada. May 2000. Presses internationales Polytechnique, Montreal, Canada, May 2000.

[30] G. Shen, T. Sakkalis and N. M. Patrikalakis, Interval Methods for B-Rep Model Verification and Rectification, *Proceedings of the ASME 26th Design Automation Conference*, Baltimore, MD, September 2000. p. 140 and CDROM. NY: ASME, 2000.

[31] N. M. Patrikalakis, T. Sakkalis and G. Shen, Boundary Representation Models: Validity and Rectification, Invited paper in *Proceedings of the 9th IMA Conference on the Mathematics of Surfaces.* University of Cambridge, UK. September 2000. R. Cipolla and R. Martin, editors. pp. 389-409. London, UK: Springer-Verlag, 2000.