HAL
open science

# Converting Hybrid Wire-frames to B-rep Models

Jie-Hui Gong, Hui Zhang, Yi-Wen Zhang, Jia-Guang Sun

## To cite this version:

Jie-Hui Gong, Hui Zhang, Yi-Wen Zhang, Jia-Guang Sun. Converting Hybrid Wire-frames to B-rep Models. SPM '07: Proceedings of the 2007 ACM symposium on Solid and physical modeling, ACM, Jun 2007, Beijing, China. pp.283-289, 10.1145/1236246.1236286 . inria-00202636

**HAL Id: inria-00202636**

**https://inria.hal.science/inria-00202636v1**

Submitted on 7 Jan 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Converting Hybrid Wire-frames to B-rep Models*

Jie-Hui Gong†
Dept. of Comp. Sci. and Technol., Tsinghua University

Hui Zhang
School of Software, Tsinghua University

Yi-Wen Zhang
School of Software, Tsinghua University

Jia-Guang Sun
Tsinghua University

## Abstract

Solid reconstruction from engineering drawings is one of the efficient technologies to product solid models. The B-rep oriented approach provides a practical way for reconstructing a wide range of objects. However, its major limitation is the computational complexity involved in the search for all valid faces from the intermediate wire-frame, especially for objects with complicated face topologies. In previous work, we presented a hint-based algorithm to recognize quadric surfaces from orthographic views and generate a hybrid wire-frame as the intermediate model of our B-rep oriented method. As a key stage in the process of solid reconstructing, we propose an algorithm to convert the hybrid wire-frame to the final B-rep model by extracting all the rest faces of planes based on graph theory. The entities lying on the same planar surface are first collected in a plane graph. After all the cycles are traced in a simplified edge-adjacency matrix of the graph, the face loops of the plane are formed by testing loop containment and assigning loop directions. Finally, the B-rep model is constructed by sewing all the plane faces based on the Möbius rule. The method can efficiently construct 2-manifold objects with a variety of face topologies, which is illustrated by results of implementation.

**CR Categories:** I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Boundary Representations; G.2.2 [Discrete Mathematics]: Graph Theory—Graph Algorithms; J.6 [Computer-Aided Engineering]: Computer-Aided Design

**Keywords:** Solid reconstruction, Model conversion, Graph theory, 2-manifold, Möbius rule.

## 1 Introduction

Nowadays, solid modelers have become popular mechanical CAD tools, and solid models are necessary for some computer-aided product development techniques, such as finite-element analysis, process planning, and numerically controlled machining. Even so, engineering drawings still plays an essential role in engineering practice as a standard language for describing mechanical designs. The information embedded in 2D drawings cannot be directly used in 3D CAD systems. Therefore, solid reconstruction from engineering drawings is one of the efficient technologies to product solid models for the maintenance of considerable legacy designs and the upgrade of existing products based on their designs.

A body of work related to the solid reconstruction exists in the literature [Nagendra and Gujar 1988; Wang and Grinstein 1993]. In terms of solid representations, existing reconstruction schemes can be basically classified into two major categories: B-rep oriented approach and CSG oriented approach. The former has one important advantage over the latter, which is reflected in the complexity of the object domain that can be reconstructed. The B-rep oriented approach can handle more complicated polyhedra and so far has extended the object domain into quadric solid models without restrictions on the axes of curved surfaces [Liu et al. 2001; Gong et al. 2006b]. The CSG oriented approach, however, is generally applicable only to mechanical parts of uniform thickness or axis-aligned volumes of revolution [Soni and Gurumoorthy 2003; Lee and Han 2005], because it typically uses either pre-defined primitives or identifies entities that can be extruded or rotated.

The B-rep oriented approach provides a practical way for automatically reconstructing mechanical parts from their orthographic views. It is based on the idea of constructing an intermediate wire-frame [Wesley and Markowsky 1981]. Therefore, all algorithms based on the approach need to solve a sub-problem, i.e. extraction of valid surfaces from the intermediate wire-frame. Previous efforts to solve this problem can be grouped into topological and geometric approaches in nature. By employing concepts of graph theory, the topological approach uses only the connectivity information of the wire-frame to derive the face topologies. Existing topological approaches can be put into three subcategories: simplicial decomposition [Agarwal and Waggenspack 1992], graph embedding [Inoue et al. 2003], and cycle space [Bagali and Waggenspack 1995]. The topological approach provides a uniform framework to handle all surface types. However, it cannot completely reconstruct the B-rep model without the geometry of the surfaces.

Almost all the B-rep oriented methods employ the geometric approach to extract faces from the intermediate wire-frame. The geometric approach determines the topology of the solid model based on the geometry of the wire-frame. It assumes that the types of two adjacent edges and the relationship between them can define a surface equation. Once the type of a surface is defined, a depth-first search is used to trace all the edges on the surface. Since a face loop is closed, the search is completed when it returns to the start vertex, provided that the edges intersect each other only at the endpoints. Tracing planar loops is very straightforward by using the vector product of two adjacent straight edges to define a plane normal, and therefore the approach is efficient for polyhedra bounded by planes [Markowsky and Wesley 1980; Yan et al. 1994].

In geometrical approaches [Sakurai and Gossard 1983; Ah-Soon and Tombre 1995; Kuo 1998], the type relationships between edges and surfaces must be sorted to comply with certain restrictions on both surfaces types and face topologies for the target curved object. A curved surface equation derived from two adjacent edges may be ill-defined since the edge types do not provide strict restriction on the type of the surface. For example, a straight edge and a circular edge could define a planar, cylindrical, conical or other higher order curved surface. On the other hand, a quadric surface could be defined by any two kinds of conic edges. Therefore, an exhaustive search must be used for revealing all valid faces within certain object domain, which is very expensive. Some recently proposed work employed accelerating techniques to reduce the searching time. Shin et al. [Shin and Shin 1998] introduced auxiliary

*This work was supported by the 973 Program of China (Grant No. 2004CB719404) and the National Nature Science Foundation of China (Grant No. 60635020).

†e-mail: gongjh03@mails.tsinghua.edu.cn

Copyright © 2007 by the Association for Computing Machinery, Inc.
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions Dept, ACM Inc., fax +1 (212) 869-0481 or e-mail permissions@acm.org.
SPM 2007, Beijing, China, June 04 – 06, 2007.
© 2007 ACM 978-1-59593-666-0/07/0006 $5.00

edges of silhouette and tangency to define cylindrical and toroidal surfaces. Kuo [Kuo 2001] employed the minimum internal angle algorithm to find quadric faces, preventing the inner-edge subdivisions and the unwanted outer loops. Liu et al. [Liu et al. 2002] used the relationship between the boundaries of faces and their projections to generate face loops. Since existing B-rep oriented methods only use low-level information of adjacency relationship between projected entities, some ghost elements that generally involve redundant and pathological edges may exist in the wire-frame. Therefore, converting wire-frame models to B-rep models still struggles in intricate cases of objects with complicated topology of faces, where long processing time is involved in the search for all valid faces bounding the object with the possibility of backtracking and heuristics.

In our previous paper [Gong et al. 2006b], an algorithm was proposed to directly recognize quadric surfaces from orthographic views by hint-based pattern matching. With that, a hybrid wire-frame has been constructed instead of a conventional wire-frame, which consists of the geometry and topology of quadric faces besides vertices and edges. In this paper, we present an algorithm to convert the hybrid wire-frame to the final B-rep model by retrieving the geometry and topology of planar faces based on topological properties of the hybrid wire-frame. The method can efficiently construct 2-manifold objects with a variety of face topologies, which satisfy the Möbius rule.

## 2 Overview of Solid Reconstruction Approach

The natural quadrics (i.e. the sphere, cylinder, cone, and plane) are by far the most commonly occurring surfaces used in modelling mechanical parts. For our B-rep oriented method, three orthographic views are used as input, and objects to be reconstructed are bounded by natural quadric faces. The whole procedure of solid reconstruction is illustrated in Fig. 1, where a hybrid wire-frame is employed as the intermediate model. In the input phase, three orthographic views of a mechanical part are input from a neutral CAD file, and geometric entities maintained in the file are considered.

In the preprocessing phase (as described in [Liu et al. 2001]), the geometric data are first checked and converted to a planar graph with distinct edges connecting only at their end vertices. Then, the edges and vertices are grouped into views according to the characteristics of three orthographic views. Finally, each view is transformed from view-based coordinate system to object-based coordinates based on the convention of engineering drawings.

Three steps involved in the reconstruction phase are as follows.

1. Incomplete wire-frame generation

   Based on the planar graphs of the views, 3D linear edges along with its two terminal vertices are generated from the match bounding boxes of its projections according to the coordinate correspondences, as described in [Gong et al. 2006a]. Thus, an incomplete wire-frame is constructed in the absence of curvilinear edges.

2. Quadric surface feature recognition

   The Link-Relation Graph (LRG) is introduced as a multigraph representation of orthographic views, and quadric surface features (QSFs) are defined by special basic patterns of LRG as well as aggregation rules. By hint-based pattern matching in the LRGs of three orthographic views in an order of priority, the corresponding QSFs are recognized, and the geometry and topology of quadric faces are recovered simultaneously. Consequently, a hybrid wire-frame is constructed by assembling the quadric faces into the incomplete wire-frame. The algorithm is described in [Gong et al. 2006b].

3. Planar surface extraction

   All the planar faces in the hybrid wire-frame are extracted based on the properties of 2-manifolds, and the B-rep solid model is reconstructed. The procedure is described in the next section.

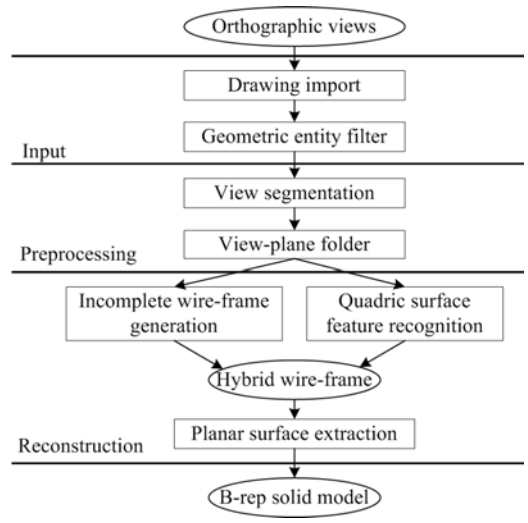



Figure 1: Pipeline of solid reconstruction employing a hybrid wire-frame.

## 3 Graph theory for plane extraction

### 3.1 Preliminaries

Most of the terminology reviewed here is the same as that used in standard graph theory [Diestel 2005].

Let $O \subseteq \mathbb{R}^2$ be an open set. Being linked by an edge in $O$ defines an equivalence relation on $O$. The corresponding equivalence classes are again open; they are the *regions* of $O$. A closed set $X \subseteq \mathbb{R}^2$ is said to separate $O$ if $O \backslash X$ has more than one region. The *frontier* of a set $X \subseteq \mathbb{R}^2$ is the set $Y$ of all points $y \in \mathbb{R}^2$ such that every neighborhood of $y$ meets both $X$ and $\mathbb{R}^2 \backslash X$.

**Property 1** *Let $X_1, X_2 \subseteq \mathbb{R}^2$ be disjoint sets, each the union of finitely many vertices and edges, and let $P$ be an edge between a vertex in $X_1$ and one in $X_2$ whose interior $\dot{P}$ lies in a region $O$ of $\mathbb{R}^2 \backslash (X_1 \bigcup X_2)$. Then $O \backslash \dot{P}$ is a region of $\mathbb{R}^2 \backslash (X_1 \bigcup P \bigcup X_2)$ [Diestel 2005].*

A *plane graph* $G$ is a pair $(V, E)$ of finite sets with the following properties (the elements of $V$ are called *vertices*; those of $E$ are *edges*):

1. $V \subseteq \mathbb{R}^2$;
2. every edge is an arc between two vertices;
3. different edges have different sets of endpoints;
4. the interior of an edge contains no vertex and no point of any other edge.

For every plane graph $G$, the set $\mathbb{R}^2 \backslash G$ is open; its regions are the *faces* of $G$.

An alternating sequence of edges is termed a *path* if all the vertices are distinct, and a closed path is called a *cycle*. An edge which joins two vertices of a cycle but is not itself an edge of the cycle is a *chord* of that cycle. While, the *bridges* in a graph are precisely those edges that do not lie on any cycle.

### 3.2 Topological Properties of Hybrid Wire-frame

A *hybrid wire-frame* $H = (V, E, Q)$ is the curvilinear wire-frame of a curved object along with the topology and geometry of all quadric surfaces, where $V$ is a set of vertices, $E$ is a set of edges, and $Q$ is a set of quadric surfaces that are unambiguously identified as a connected set of edges, characterized by the type of surface geometry with geometric parameters.

According to the *Möbius rule*, every face in the B-rep model is oriented by an associated "outward normal" pointing toward the outside of the object at any point on the face. The outer loop of the face is traversed in an anticlockwise direction as seen from the outside of the face. In addition, nested loops belonging to the same face are traversed in opposite directions by the *nesting rule* [Vosniakos 1998].

For the practicality of manufacturing, a *2-manifold* object is a solid where every point on its surface has a neighborhood topology equivalent to an open disk in $\mathbb{R}^2$. The B-rep model of a 2-manifold satisfies the following properties [Mantyla 1986].

**Property 2** *Every edge belongs to exactly two faces, and the orientation of the edge is inverted by each face.*

**Property 3** *Faces may not intersect each other except at common edges or vertices.*

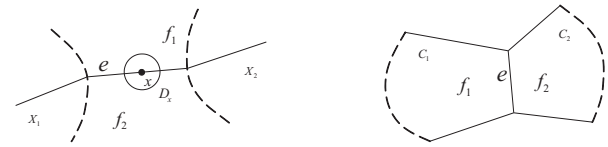
Consequently, in the hybrid wire-frame representing the manifold, all the vertices and edges on the same planar surface can be collected as a plane graph [Shin and Shin 1998].

**Property 4** *Let $G$ be a plane graph containing all the vertices and edges on the same plane in the hybrid wire-frame that represents a 2-manifold object. Then every edge in $G$ must be and only be on one cycle of $G$.*

*Proof.* Let $e$ be an edge of $G$. Referring to Figure 2(a), around every point $x \in \dot{e}$ ($\dot{e}$ is the interior of $e$) we can find an open disc $D_x$ with center $x$, which meets only $e$. Then $D_x \backslash G$ is the union of two open half-discs, which do not meet $G$, i.e. $(D_x \backslash G) \bigcap G = \emptyset$. Thus, each of the half-discs lies in a face of $G$. Let us denote these faces by $f_1$ and $f_2$; they are the only faces of $G$ with $x$ on their frontier, and they may coincide. Consequently, $e$ must be on the coincident frontier of $f_1$ and $f_2$.

If $e$ does not lie on a cycle, then $e$ is a bridge and thus links two disjoint subgraphs $X_1$ and $X_2$ as in Property 1, with $X_1 \bigcup X_2 = G \backslash \dot{e}$. Let $f$ be the face of $G - e$, i.e. $f = \mathbb{R}^2 \backslash (X_1 \bigcup X_2) = \mathbb{R}^2 \backslash (G - e)$. Clearly, $f_1 \bigcup \dot{e} \bigcup f_2$ is a subset of $f$. By Property 1, $f \backslash \dot{e}$ is a face of $G$, i.e. $f \backslash \dot{e} = \mathbb{R}^2 \backslash (X_1 \bigcup e \bigcup X_2)$. But $f \backslash \dot{e}$ contains $f_1$ and $f_2$ by the definition of $f$, so $f_1 = f = f_2$ since $f_1$, $f_2$, and $f$ are all faces of $G$. Consequently, $e$ does not separate the region of $\mathbb{R}^2 \backslash (G - e)$. This is in contradiction with Property 2. Therefore, every edge of $G$ must be on one cycle of $G$.

If $e$ belongs to two cycles of $G$, denoted as $C_1$ and $C_2$ in Figure 2(b), then $e$ is a chord of the cycle $C_1 \bigcup C_2$. $f_1$ and $f_2$ are two distinct faces of $G$, which are bounded by $C_1$ and $C_2$, respectively. But $C_1$ and $C_2$ are on the same planar surface due to the definition of $G$, which means that two adjacent faces sharing a common edge are coplanar. This contradicts Property 3. Similarly, the common vertex shared by two cycles also induces the same contradiction. Therefore, every edge of $G$ must be on only one cycle of $G$. □




(a) Edge $e$ is a bridge. (b) Edge $e$ is a chord.

Figure 2: Edges in $G$.

**Property 5** *All the entities within $G$ form several disjoint cycles.*

*Remark.* According to Property 4, every edge in $G$ must belongs to distinct cycles of $G$. Therefore, $G$ contains several cycles without sharing edges and vertices, i.e. disjoint cycles. □

## 4 Plane Extracting Algorithm

The main task for converting the hybrid wire-frame to the B-rep model is to retrieve the geometry and topology of planar surfaces. This stage begins with the hybrid wire-frame model generated in the previous stage. First, all the planar surfaces are found within the hybrid wire-frame, with all the entities lying on the same surface collected onto a plane graph. Then, all the cycles are traced in the graph for each planar surface. Subsequently, the outer loop and inner loops are sorted by the containment test and assigned respective directions according to the nesting rule. Finally, the B-rep model is constructed by checking the direction consistency of faces based on the Möbius rule. These steps are described in the following subsections.

### 4.1 Generating Plane Graphs

The hybrid wire-frame has contained the topological information of all the quadric surfaces on the boundary of the object, thus all the edges in the hybrid wire-frame can be classified according to the *cycle-incidence number* of an edge $e$, denoted as $D(e)$. By Property 2, there are three types of edges: (1) $D(e) = 0$ means $e$ incident to no cycle; the edge is shared by two planar faces. (2) $D(e) = 1$ hints that $e$ belongs to a cycle of planar face due to associating itself with a loop of certain quadric face. (3) $D(e) = 2$ implies the edge has been incident with loops of two incident quadric faces.

It is very straightforward to define a plane normal by using the vector product of two adjacent straight edges [Kuo 2001]. In addition, a plane equation is derived from a straight edge and a conic edge, as well as one closed conic.

Adjacency matrices can be used to represent graph structures [Diestel 2005]. According to Property 5, every edge within the plane graph is adjacent to two distinct edges at its two end-vertices, respectively, distinguishing a closed edge as a cycle itself. Therefore, a *simplified adjacency matrix* $A = (a_{ij})_{n \times 2}$ of edges in $G$, where $n$ is the number of edges, is defined as below.

$$a_{ij} = \begin{cases} k & e_k \in E; \\ -1 & e_i \text{ is closed conic.} \end{cases}$$

Every row vector $a_i$ of $A$ indexes adjacency edges of $e_i$, with $a_{i0}$ for the edge at the start vertex of $e_i$, and $a_{i1}$ for the other one.

The *un-sewed* edges are those of $D(e) \neq 2$ in the hybrid wire-frame. Figure 3 shows the procedure for finding all the planar surfaces according to the un-sewed edges. Consequently, all the entities lying

on the same planar surface are collected as the corresponding plane graph.

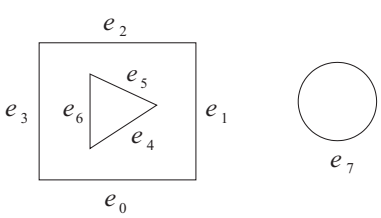
```
GeneratePlaneGraphs(W_H, F_G)
/* W_H: hybrid wire-frame; F_G: set of plane graphs.*/
  For (each edge e_i in W_H) do
    If (D(e_i) == 0) then
      For (each edge e_j adjacent to e_i) do
        If (D(e_j) == 0) then
          Define the plane equation f according to e_i and e_j.
          Call ConstructOneGraph(f, G).
          Add e_i and e_j into G.
    Else if (D(e_i) == 1) then
      If (e_i is a closed conic) then
        Define f according to e_i.
        Call ConstructOneGraph(f, G).
        Add e_i into G.
      Else if (e_i is linear) then
        For (each edge e_j adjacent to e_i) do
          If (D(e_j) == 1) then
            Define f according to e_i and e_j.
            Call ConstructOneGraph(f, G).
            Add e_i and e_j into G.

ConstructOneGraph(f, G)
/* f: plane equation; G: plane graph. */
  For (each plane graph G_i in F_G) do
    If (G_i lies on the plane defined by f) then
      G ← G_i, and return.
  Create a plane graph G based on f.
  Add G into F_G, and return.
```

Figure 3: Pseudo-code for generating plane graphs of a hybrid wire-frame.

## 4.2 Tracing Cycles

For each planar surface sought in the previous step, all the edges within its plane graph are indexed by the adjacency matrix, which is illustrated with Figure 4.




(a) $G$

| | | |
|---|---|---|
| $e_0$ | 3 | 1 |
| $e_1$ | 0 | 2 |
| $e_2$ | 3 | 1 |
| $e_3$ | 0 | 2 |
| $e_4$ | 6 | 5 |
| $e_5$ | 4 | 6 |
| $e_6$ | 4 | 5 |
| $e_7$ | -1 | -1 |

(b) $A$

Figure 4: Topological structure of plane graph $G$ is described by the simplified edge-adjacency matrix $A$.

In this step, based on the adjacency matrix, all the cycles are traced to form the boundary of the surface. The exploration is a depth-first search, which starts with an un-sewed edge, until it is back to the starting edge. The algorithm employed to detect cycles in a plane graph is shown in Figure 5.

Therefore, by tracing adjacent relations of edges recorded in the adjacency matrix (Figure 4(b)), three cycles in the plane graph (Figure 4(a)) are detected:

$c_0 : e_7 \rightarrow e_7$;

```
TraceCycles(A, F_c)
/*A: simplified adjacency matrix; F_c: set of cycles.*/
  For (each vector a_i of A) do
    If (a_i0 == −1 and a_i1 == −1) then
      Take e_i as a cycle c, and add c into F_c.
  While (there is a edge e_i in A, with a_i0 ≠ −1 or a_i1 ≠ −1)
    Construct a cycle c starting with e_i, and e_s ← e_i.
    Select the direction d of e_s for travelling.
    While (until e_s appears again)
      Find the edge e_j adjacent to e_i along d, based on a_i.
      Add e_j into c.
      Label a_i and a_j referring to d.
      e_j ← e_i.
    Add c into F_c.
```

Figure 5: Pseudo-code for tracing cycles in a plane graph.

$c_1 : e_0 \rightarrow e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow e_0$;
$c_2 : e_4 \rightarrow e_5 \rightarrow e_6 \rightarrow e_4$.

## 4.3 Forming Face Loops

In this step, all the cycles within the graph are first sorted into inner ones and outer ones by containment testing and then assigned directions to form face loops.

Firstly, the containment relationships of cycles are retrieved by checking all the cycles pairwise for possible nesting. By Property 5, one cycle must be either outside (*isolated*) or inside (*nested*) the other cycle, without any common elements. Therefore, the containment relationship of two cycles can be determined by proving that just one point of the nested cycle is inside the nesting cycle. The *ray-casting method* [Vosniakos 1998] is used for testing cycle containment. Frequently, there are several cycles with isolated and nested relations in a hierarchical manner, which can be revealed by the data structure shown in Figure 6. For each cycle $c$, an *outer container* $O(c)$ is employed to record the cycles containing $c$ and an *inner container* $I(c)$ for the ones nested by $c$.

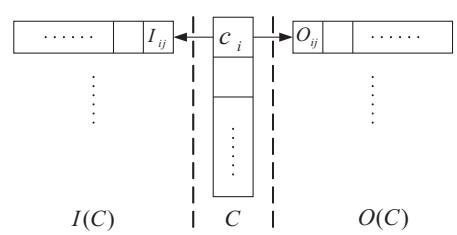



Figure 6: Containment relationships of cycles $C$ in a plane graph are recorded by an inner container $I(C)$ and an outer container $O(C)$.

Then, nested cycles are directed by reference to the plane normal. The faces on the same plane surface are defined by forming nested loops in pairs of cycle nesting levels. The procedure for forming loops of faces is given in Figure 7.

Finally, based on the Möbius rule, all the plane faces are sewed into the hybrid wire-frame by identifying the orientation consistency of faces for the B-rep model. A face adjacent to a vertex with minimum $x$ coordinate is used to decide which side is outside. From this, all other faces are oriented by comparing the direction of traversal of their edge loops. The sign of a face normal is reversed if adjacent face loops traverse adjacent edges in the same direction. Once the status of one face has been established, the remaining can be deduced.

```
DefineFaces(F_c, I, O, F_p)
/*F_c: set of cycles; I: inner container; O: outer container;
  F_p: set of faces.*/
  For (each cycle c_i in F_c) do
    If (there are even numbers of cycles in O(c_i)) then
      Label c_i as a outer loop and add into the list L_o.
      For(each inner container I_j in I, with j ≠ i) do
        For(each cycle c_k in I_j) do
          If (c_k is the same as c_i) then
            Delete c_k from I_j. /* no multi-nested loops. */
  For (each outer loop l_i in L_o) do
    Define a face P taking l_i as the outer loop.
    Assign the anticlockwise direction for l_i.
    For (each cycle c_j in the inner container of l_i) do
      Add c_j in P as a inner loop l_j.
      Assign the clockwise direction for l_j.
    Add P into F_p.
```

Figure 7: Pseudo-code for defining faces based on nested cycles.

## 5 Implementation

The implementation of the proposed conversion algorithm was tested as the last stage of the reconstruction module in the commercial CAD system TiGEMS® 6.0 that is possessed by Tsinghua Software Information Technology Co. Ltd. The developed program accepts standard drawing files in DXF format that consist of three orthographic views. The hybrid wire-frames were generated as intermediate models consisting of all the edges and quadric faces, and the B-rep models of the corresponding 2-manifolds were constructed after retrieving and sewing all the planar faces. Six real mechanical parts are provided, in Figures 8–13, to demonstrate various cases that can be handled by our method. They were tested on a personal computer platform with an Intel Pentium D 2.80GHz CPU and 448MB RAM. Resulting models are rendered in order to confirm that the proposed algorithm can construct solid models adequately.

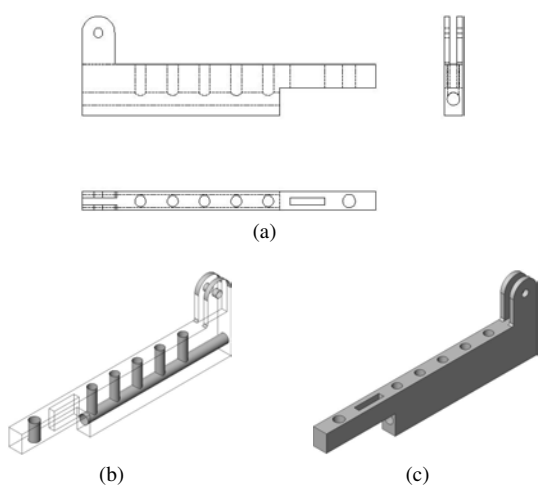



Figure 8: Example 1. (a) The three orthographic views. (b) There are 67 un-sewed edges in the hybrid wire-frame. (c) There are 13 faces of quadrics and 16 faces of planes in the B-rep model.

Figure 8 shows a mechanical part with a linear array of holes. The process of converting the hybrid wire-frame (Figure 8(b)) to the B-rep models (Figure 8(c)) occurred in 5 *ms*. Figure 9 is an example involving multiple adjacency between planes and quadrics.

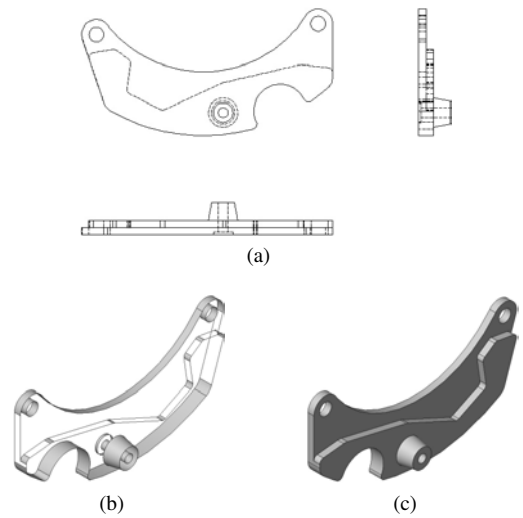



Figure 9: Example 2. (a) The three orthographic views. (b) The hybrid wire-frame involves 23 faces of quadrics, as well as 86 un-sewed edges to be traced. (c) Fourteen planar faces were extracted to form the B-rep model.

The B-rep model in Figure 9(c) was constructed in 6 *ms*, by travelling un-sewed edges in the hybrid wire-frame (Figure 9(b)). An example whose shape varies rapidly is shown in Figure 10. The hybrid wire-frame in Figure 10(b) was converted to the B-rep model (Figure 10(c)) in 9 *ms*. Figure 11 gives the result of a mechanical part with complex exterior. Based on the hybrid wire-frame in Figure 11(b), the processing time for the model conversion is about 12 *ms*, which results in the B-rep model in Figure 11(c). In Figure 12, almost all the outer surfaces of the mechanical part are planes. Figure 12(b) is the hybrid wire-frame to be converted, and Figure 12(c) is the final B-rep model constructed in 12 *ms*. The mechanical part provided in Figure 13 possesses the complex topology of faces that arise from several blends. The B-rep model (Figure 13(c)) was reconstructed by tracing all planar surfaces in the hybrid wire-frame (Figure 13(b)). The processing time is about 20 *ms*.

Table 1 summaries the experimental results applied to these examples. For the hybrid wire-frames, the un-sewed edges of $D(e) = 0$ and $D(e) = 1$ are counted respectively. In addition, beside all the edges in the B-rep model, the quadric faces and planar faces are also separately taken count. Finally, the CPU times are provided for converting the hybrid wire-frames to the B-rep models. Based on the examples, the following observations can be made about the conversion method.

- All the examples were successively constructed in less than 0.1 second of CPU time. The running time for sewing a B-rep model is proportional to the number of un-sewed edges in a hybrid wire-frame but independent of the number of planar faces to be traced. Therefore, the proposed algorithm is efficient enough to on-line serve B-rep models from hybrid wire-frames for the reconstruction of objects with moderate structure complexity.
- The examples exhibit a great diversity of face topologies in the B-rep models. Based on the same number of un-sewed edges in a hybrid wire-frame, the less planar faces extracted, the higher the complexity of face topologies formed in a B-rep model. Therefore, the presented algorithm can applicable to common mechanical parts, whose shapes are expressed by various face topologies.

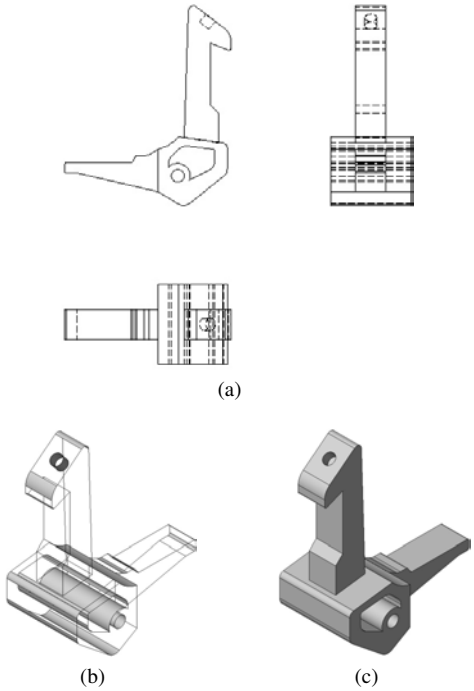



Figure 10: Example 3. (a) The three orthographic views. (b) The hybrid wire-frame contains 133 un-sewed edges. (c) There are 19 quadric faces and 21 planar faces in the B-rep model.

Table 1: Performance evalution of examples

| Example | $N(E_0)$ | $N(E_1)$ | $N(E)$ | $N(Q)$ | $N(P)$ | Time* |
|---|---|---|---|---|---|---|
| Figure 8 | 38 | 29 | 81 | 13 | 16 | 5 |
| Figure 9 | 20 | 66 | 99 | 23 | 14 | 6 |
| Figure 10 | 61 | 72 | 135 | 19 | 31 | 9 |
| Figure 11 | 64 | 84 | 165 | 29 | 28 | 12 |
| Figure 12 | 71 | 84 | 171 | 29 | 47 | 12 |
| Figure 13 | 100 | 148 | 290 | 58 | 57 | 20 |

$N(X)$: number of elements in set $X$.

$*$: CPU time in *ms* on a personal computer platform with an Intel Pentium D 2.80GHz CPU and 448MB RAM.

## 6 Conclusion

An approach is proposed to convert hybrid wire-frame models to B-rep models, after constructing quadric surfaces from three orthographic views by the hint-based feature recognition as described in [Gong et al. 2006b]. The algorithm employs a graph-based technique to recover the geometry and topology of planar surfaces from the hybrid wire-frame, and therefore the B-rep model of a 2-manifold object is completed based on the Möbius rule. By the algorithm, B-rep models can be on-line served from hybrid wire-frames for the solid reconstruction. Furthermore, the algorithm can accommodate to mechanical parts with various topologies of faces. The implementation of the algorithm was verified using a number of sample parts, and several illustrative examples are provided to demonstrate the variety of cases that can be efficiently handled.

The proposed algorithm is based on perfect hybrid wire-frame models. However, sometimes it is difficult to ensure that the drawing is accurate and consistent, which may induce ghost elements in the hybrid wire-frames. Therefore, techniques for reconstructing 3D objects from real drawings are necessary to take imperfections into account.

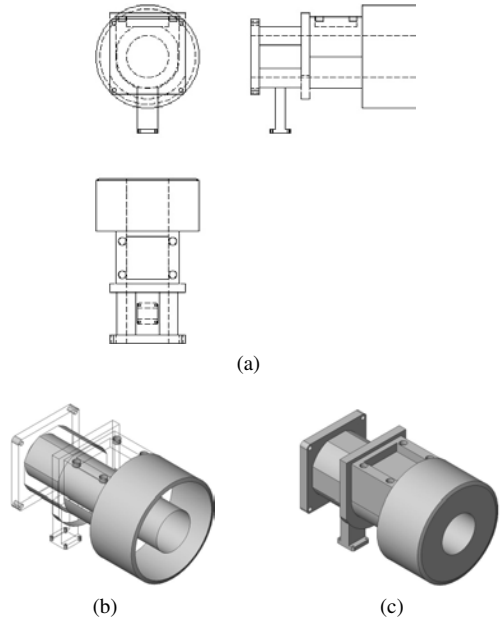



Figure 11: Example 4. (a) The three orthographic views. (b) There are 148 un-sewed edges involved in the hybrid wire-frame. (c) The completed B-rep model consists of 67 faces, where are 29 quadrics and 38 planes.

## References


AGARWAL, S. C., AND WAGGENSPACK, W. N. J. 1992. Decomposition method for extracting face topologies from wireframe models. *Computer-Aided Design 24*, 3, 123–140.

AH-SOON, C., AND TOMBRE, K. 1995. A step towards reconstruction of 3-d cad models from engineering drawings. In *Proceedings of the Third International Conference on Document Analysis and Recognition*, 331–334.

BAGALI, S., AND WAGGENSPACK, W. N. J. 1995. A shortest path approach to wireframe to solid model conversion. In *Proceedings of the third ACM Symposium on Solid Modeling and Applications*, 339–350.

DIESTEL, R. 2005. *Graph theory (third edition)*. Berlin: Springer-Verlag.

GONG, J. H., ZHANG, G. F., ZHANG, H., AND SUN, J. G. 2006. Reconstruction of 3d curvilinear wire-frame from three orthographic views. *Computer and Graphics 30*, 2, 213–224.

GONG, J. H., ZHANG, H., ZHANG, G. F., AND SUN, J. G. 2006. Solid reconstruction using recognition of quadric surfaces from orthographic views. *Computer-Aided Design 38*, 8, 821–835.

INOUE, K., SHIMADA, K., AND CHILAKA, K. 2003. Solid model reconstruction of wireframe cad models based on topological embeddings of planar graphs. *Journal of Mechanical Design*, 125, 434–442.

KUO, M. H. 1998. Reconstruction of quadric surface solids from three-view engineering drawings. *Computer-Aided Design 30*, 7, 517–527.

KUO, M. H. 2001. Automatic extraction of quadric surfaces from wire-frame models. *Computers and Graphics 25*, 1, 109–119.

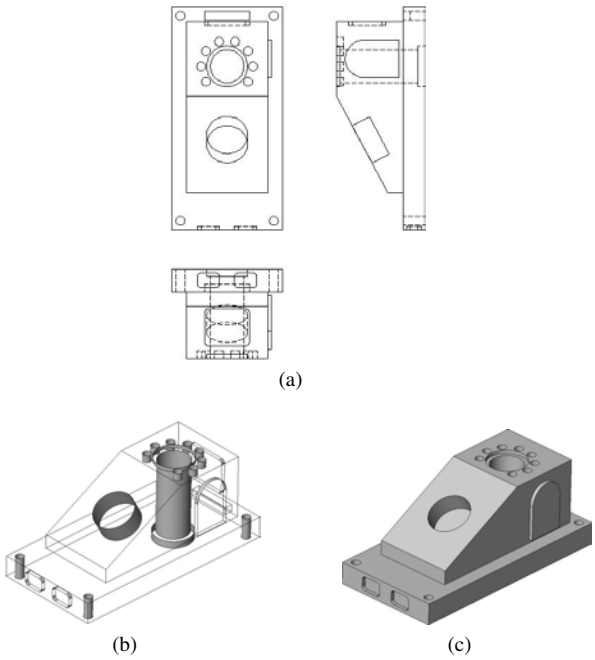



Figure 12: Example 5. (a) The three orthographic views. (b) The hybrid wire-frame consists of 155 un-sewed edges, besides 16 edges shared by 29 quadric faces. (c) The B-rep model was reconstructed by sewing another 47 plane faces.

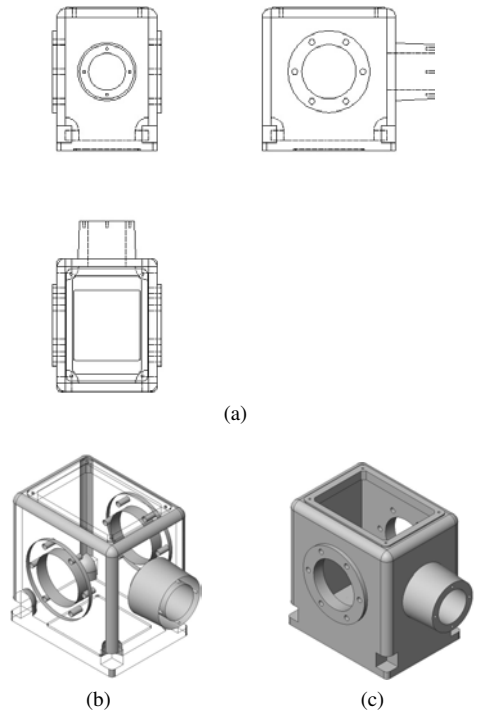



Figure 13: Example 6. (a) The three orthographic views. (b) The hybrid wire-frame includes 248 un-sewed edges. (c) The reconstructed B-rep model contains 290 edges and 115 faces, including 58 quadrics and 57 planes.


LEE, H., AND HAN, S. 2005. Reconstruction of 3d interacting solids of revolution from 2d orthographic views. *Computer-Aided Design 37*, 13, 1388–1398.

LIU, S. X., HU, S. M., CHEN, Y. J., AND SUN, J. G. 2001. Reconstruction of curved solids from engineering drawings. *Computer-Aided Design 33*, 14, 1059–1072.

LIU, S. X., HU, S. M., AND SUN, J. G. 2002. Two accelerating techniques for 3d reconstruction. *Journal of Computer Science and Technology 17*, 3, 362–368.

MANTYLA, M. 1986. Boolean operations of 2-manifolds through vertex neighborhood classification. *ACM Transactions on Graphics 5*, 1, 1–29.

MARKOWSKY, G., AND WESLEY, M. A. 1980. Fleshing out wire frames. *IBM Journal of Research and Development 24*, 5, 582–597.

NAGENDRA, I. V., AND GUJAR, U. G. 1988. 3-d objects from 2-d orthographic views – a survey. *Computers and Graphics 12*, 1, 111–114.

SAKURAI, H., AND GOSSARD, D. C. 1983. Solid model input through orthographic views. *ACM/SIGGRAPH Computer Graphics 17*, 3, 243–252.

SHIN, B. S., AND SHIN, Y. G. 1998. Fast 3d solid model reconstruction from orthogrpahic views. *Computer-Aided Design 30*, 1, 63–76.

SONI, S., AND GURUMOORTHY, B. 2003. Handling solids of revolution in volume-based construction of solid models from orthographic views. *Journal of Computing and Information Science in Engineering 3*, 250–259.

VOSNIAKOS, G. 1998. Conversion of wireframe to acis solid models for $2\frac{1}{2}$ d engineering components. *Advanced Manufacturing Technology 14*, 199–209.

WANG, W., AND GRINSTEIN, G. G. 1993. A survey of 3d solid reconstruction from 2d projection line drawings. *Computer Graphics Forum 12*, 2, 137–158.

WESLEY, M. A., AND MARKOWSKY, G. 1981. Fleshing out projections. *IBM Journal of Research and Development 25*, 6, 934–953.

YAN, Q. W., CHEN, C. L. P., AND TANG, Z. S. 1994. Efficient algorithm for the reconstruction of 3d objects from orthographic projections. *Computer-Aided Design 26*, 9, 699–717.