Reconstruction of curved solids from engineering drawings

Shi-Xia Liu*, Shi-Min Hu, Yu-Jian Chen, Jia-Guang Sun

Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

Abstract

This paper presents a new approach for reconstructing solids with planar, quadric and toroidal surfaces from three-view engineering drawings. By applying geometric theory to 3-D reconstruction, our method is able to remove restrictions placed on the axes of curved surfaces by existing methods. The main feature of our algorithm is that it combines the geometric properties of conics with affine properties to recover a wider range of 3-D edges. First, the algorithm determines the type of each 3-D candidate conic edge based on its projections in three orthographic views, and then generates that candidate edge using the conjugate diameter method. This step produces a wire-frame model that contains all candidate vertices and candidate edges. Next, a maximum turning angle method is developed to find all the candidate faces in the wire-frame model. Finally, a general and efficient searching technique is proposed for finding valid solids from the candidate faces; the technique greatly reduces the searching space and the backtracking incidents. Several examples are given to demonstrate the efficiency and capability of the proposed algorithm. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords: Engineering drawings; Wire-frame; Solid modeling; Reconstruction

1. Introduction

Rapid changes in CAD technologies have left behind a huge quantity of legacy data in the form of engineering drawings. Many of these older designs are needed for maintaining existing mechanical systems and as a basis for designing the next generation of products. 2-D descriptions of engineering drawings are unintuitive, difficult to manipulate and redesign, and do not serve as a basis for computer aided design. It is therefore important to have efficient algorithms for reconstructing solid objects from the drawings. One of the earliest such algorithms was proposed by Idesawa [1].

Over the past two decades, much effort has gone into the automatic reconstruction of 3-D solid models from engineering drawings [2–4]. Basically, the reconstruction algorithms can be categorized into two general approaches according to the solid representation schemes used. The first family of algorithms, known as CSG oriented approach, adopts a top-down reconstruction strategy; while the other family, known as B-rep oriented approach, works in bottom-up fashion.

The CSG oriented approach assumes that each 3-D object can be built from certain primitives in a hierarchical manner. It selects a 2-D loop as a base and uses translational sweep operation to construct a 3-D primitive; the constructed primitives are then assembled to form the final object. This is the basic approach adopted by Chen [5], Meeran [6], Masuda [7] and Shum [8]. A main drawback of the CSG oriented approach is that it is difficult to recognize special patterns in the primitives from complicated drawings; thus it is generally applicable only to drawings of mechanical parts of uniform thickness or rotational objects.

The B-rep oriented approach, which is based on the 'fleshing out projections' concept, was formalized by Wesley and Markowsky [9,10]. The approach involves the following main steps:

1. Generate 3-D candidate vertices from 2-D vertices
2. Generate 3-D candidate edges from 3-D candidate vertices
3. Construct candidate faces from 3-D candidate edges on the same surface
4. Construct 3-D object from candidate faces

Many researchers have implemented variants of this approach. Earlier efforts of reconstruction were aimed at generating polyhedral objects from their projections [1,9,10]. Most of the later research has focused on extending the range of objects to be reconstructed. An approximation method that introduces two kinds of supplementary lines, silhouette lines and tangency lines, has been proposed to reconstruct objects with curved surfaces [11]. In order to

* Corresponding author. Tel.: +86-10-6278-2052; fax: +86-10-6278-2025.

E-mail address: liusx@ncc.cs.tsinghua.edu.cn (S.-X. Liu).

generate tangency edges and silhouette edges, Sakurai [11], Lequette [12] and Shin [13] also introduced two kinds of 2-D auxilliary vertices—silhouette vertices and tangency vertices. However, their methods may fail when there is more than one 2-D edges between two 2-D vertices in a view. Kuo [14,15] proposed a five-point method to deal with this problem. To find a one-to-one correspondence among the different views, Kuo's method has to divide all the conics into quadrants using the center lines, obtain the 3-D parametric equation for each quadrant, then combine them using one parametric equation, and finally transform the parametric form to its standard form. This is a very time-consuming process. More recently, Liu [16] and Oh [17] extended the earlier methods to deal more efficiently with curved surfaces. Tanaka [18] gave a method for automatically decomposing a 2-D assembly drawing into 3-D part drawings by a system of solid element equations.

Two major limitations in existing reconstruction methods are as follows:

1. Narrow domain of applicability due to the restriction on the axes of the curved surfaces and conics to be parallel to one of the co-ordinate axes.
2. Long processing time is needed due to combinatorial searches and complicated geometric operations. For example, in order to find solids from the reconstructed wire frame, methods based on B-rep approach have to introduce cutting edges, to check all possible combinations of 3-D subparts, and to assemble those constructed subparts to obtain all objects. This leads to an exponential growth in complexity.

This paper proposes a stable reconstruction algorithm that places no restrictions on the axes of curved surfaces and conics. To the best of our knowledge, there is no such algorithm in the literature. We have successfully overcomed the two limitations by using the conjugate diameter method to reconstruct 3-D objects with quadric and toroidal surfaces directly without using polyhedron approximation. By not introducing supplementary lines that are needed in polyhedron approximation, we manage to increase both the efficiency and reliability of the reconstruction. We have chosen to base our reconstruction algorithm on the B-Rep oriented approach because it can handle complicated drawings more easily than the CSG oriented approach. Three orthographic views are used as input. They contain two types of line segments: solid and dashed. The objects to be reconstructed are limited to compact, oriented, and 2-manifold objects, which are assumed to be bounded by only planes, quadric surfaces, or toroidal surfaces. Our reconstruction algorithm consists of the following steps:

1. Preprocess the input data for reconstruction
2. Generate a 3-D wire-frame that consists of candidate vertices and candidate conic edges from 2-D geometric elements

3. Search for 3-D candidate faces in the wire-frame model
4. Search for all possible solids from the candidate faces

Step 2 is the most crucial in this algorithm; it determines the domain of solid objects to be reconstructed. We have developed a new approach for this step based on the concept of conjugate diameters. The mathematical definition and theory involved in developing this new approach are presented in Section 2. Detailed description of the four steps in the reconstruction algorithm is described in Section 3. Section 4 shows some reconstruction results. Finally, conclusion and future work are presented in Section 5.

2. Mathematical theory for generating wire-frame

We first introduce some preliminary definitions which are useful for subsequent discussions.

Definition 1. (non-degenerate parallel projection). Under parallel projection, if the plane containing a space conic is not perpendicular to the projection plane, then the parallel projection is said to be non-degenerate.

Definition 2. (affine map). A map Φ that maps E^3 onto itself is called an affine map if it leaves the barycentric combinations invariant; that is, if

$$x = \sum \partial_j a_j \, x, a_j \in E^3 \tag{1}$$

and Φ is an affine map, then

$$\Phi x = \sum \partial_j \Phi a_j \, \Phi x, \Phi a_j \in E^3 \tag{2}$$

Definition 3. (field). Under an affine map, the set that contains all the points and all the line segments in a plane domain is called a field.

Definition 4. (corresponding points). Under an affine map, if a point $\mathbf{A}$ in a plane domain is mapped to a point $\mathbf{A}'$ in another plane domain, then the points $\mathbf{A}$ and $\mathbf{A}'$ are called corresponding points.

Definition 5. (corresponding axis) Under an affine map, the line on which every point is mapped back to the line itself is called the corresponding axis of the affine map.

Definition 6. (principal direction). If two perpendicular lines in one field correspond to two perpendicular lines in another field under an affine map, then the directions of the two lines in each field are called the principal directions of that field.
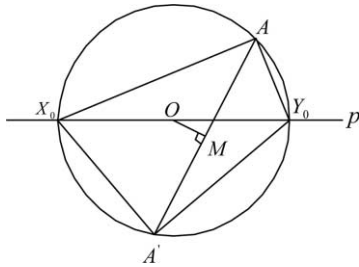
Fig. 1. The determination of the principal direction.

Definition 7. (conjugate diameter, conjugate chord). If two diameters of a central conic section are such that one of them bisects all chords that are parallel to the other, then they are called conjugate diameters. For parabolas, a diameter that bisects all chords that are parallel to each other is called a conjugate diameter, and the chords are called conjugate chords.

We shall now prove the projective property of conics, which is useful for determining the classes of space conics from their projections.

Theorem 1. If at least one of the projections of a planar curve in space is a conic, then the planar curve is also a conic.

Proof of Theorem 1. The proof is derived by contradiction and the fact that the number of times that a line intersects an algebraic curve is invariant under non-degenerate parallel projections. Suppose the degree of the planar curve in space is m, where $m \neq 2$. Without loss of generality, let the projection in the front view be non-degenerate. Suppose this projection curve is a conic. From the well-known fundamental theorem of algebra, it follows that a line intersects a curve of degree n that does not contain it, counting multiplicities at most n times [19]. Conversely, if a planar curve intersects a line at most n times, the degree of the curve is n. Since the number of times that a line intersects an algebraic curve is invariant under non-degenerate parallel projections, the projection of the space curve of degree m in the front view is therefore another algebraic curve of the same degree. This leads to a contradiction.

It follows from Theorem 1 that a projected conic corresponds to a conic in space. In addition, the class of conic curves is invariant under non-degenerate parallel projection [20]. Thus, projected ellipses, parabolas and hyperbolas correspond to space ellipses, parabolas and hyperbolas, respectively, under non-degenerate parallel projections. Therefore we can determine the class of a space conic from the class of its projections by the above property.

Since parallel projection is a special case of affine map [20], without loss of generality, we will use affine map instead of parallel projection in the following discussion.

Now we shall prove that two conjugate diameters are sufficient to determine a unique conic section. We begin with a preliminary lemma.

Lemma 1. The two pairs of principal directions of two corresponding fields under an affine map are determined by the corresponding axis of the affine map and a pair of corresponding points.

Proof. Consider two corresponding points A, A', and the corresponding axis p of an affine map, arranged as in Fig. 1. In this figure, the perpendicular bisector MO of AA' intersects axis p at point O. By the principle of perpendicular bisector, we have $OA = OA'$. Now draw a circle centered at O with radii OA, which intersects p at points X_0 and Y_0. From the definition of affine maps, we conclude that AX_0 and AY_0 correspond to $A'X_0$ and $A'Y_0$, respectively. Accordingly, since $AX_0 \perp AY_0$, $A'X_0 \perp A'Y_0$, we obtain that $AX_0 \perp AY_0$ corresponds to $A'X_0 \perp A'Y$. It follows from the above derivation that AX_0 and AY_0 are the principal directions in one field and that $A'X_0$ and $A'Y_0$ are the principal directions of the other corresponding field.

With all these preparations, we can now prove our main results.

Theorem 2. A central conic section is uniquely determined by two conjugate diameters.

Proof. Without loss of generality, we take an ellipse as an example. Since an ellipse is defined by the center point, major axis, and minor axis [21], the ellipse is therefore uniquely determined if these geometric parameters can be found from the given conjugate diameters.

Given two conjugate diameters (see AB and CD in Fig. 2), it is clear that the center of an ellipse is the intersection point of the two conjugate diameters. Thus we only need to determine the major axis and the minor axis of the ellipse. Since an affine map maps two conjugate diameters of an ellipse to two mutually perpendicular diameters of a circle, it suffices to determine the affine image (i.e., circle) of the ellipse from
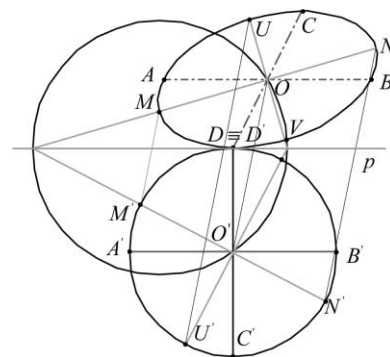

Fig. 2. Determine an ellipse from two conjugate diameters.

the conjugate diameters and find the inverse map for every point on the circle to determine the ellipse.

Referring to Fig. 2, let AB and CD be two given conjugate diameters, and let the corresponding axis of an affine map be p, which is through the endpoint D of the conjugate diameter CD and parallel to the other conjugate diameter AB. Let O be the intersection point of AB and CD. From the definition of affine maps, we obtain two perpendicular diameters $A'B'$ and $C'D'$ of a circle which correspond to the conjugate diameters AB and CD under the affine map. The corresponding point O' of O is then the intersection of $A'B'$ and $C'D'$. With the pair of corresponding points O and O' and the corresponding axis p, by Lemma 1, the principal directions MN and UV of the affine map can be determined. Since MN and UV pass through the center point O of the ellipse and are perpendicular to each other, they are the major axis and minor axis of the ellipse, respectively.

By a similar argument, we can determine a hyperbola from two conjugate diameters.

Corollary 1. A parabola is uniquely determined by a conjugate diameter and one of its conjugate chords.

Hereafter we shall use the term two conjugate diameters to encompass the case of one conjugate diameters and a chord as well.

The proof of Theorem 2 illustrates a new technique for generating 3-D candidate conic edges from three projection views. Now we give a theorem on the relationship of conjugate diameters of space conic and conjugate diameters of the projection conic.

Lemma 2. The affine ratio [20] of three points on a line is invariant under affine maps.

Referring to Fig. 3, the three points A, B, and C are collinear. Point B divides the segment AC in the ratio $(ABC) = AB/CB$. Assuming A', B', and C' are the images of the points A, B, and C, respectively, under an affine map, and that point B' divides the segment $A'C'$ in the ratio $(A'B'C') = \dfrac{A'B'}{C'B'}$. By lemma 2, we then obtain $(ABC) = (A'B'C')$.

If we can determine two conjugate diameters of a space conic from its projection conics, then by Theorem 2, the candidate space conic is uniquely determined.

Theorem 3. Two mutually perpendicular conjugate diameters of a projection conic c_p are the projections of two conjugate diameters of its corresponding space conic c.

Proof. It follows from the properties of parallel projections that the center of a projection conic c_p corresponds to the center of the space conic c, and a diameter of c_p corresponds to a diameter of c. Hence, two mutually
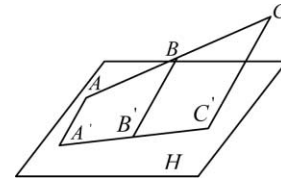

Fig. 3. Affine ratio.

perpendicular conjugate diameters of the projection conic c_p correspond respectively to two diameters of the space conic c. By Lemma 2 and the definition of conjugate diameter, we can conclude that each of the two diameters of c bisects chords that are parallel to the other diameter. Hence they are conjugate diameters. The proof is therefore completed.

Corollary 2. Two mutually perpendicular conjugate diameter and chord of a projection parabola are the projections of a conjugate diameter and a chord of its corresponding space parabola.

3. Reconstruction algorithm

In this section we describe the four steps of our reconstruction algorithm in detail.

3.1. Preprocessing 2-D data

In practice, three-view engineering drawings can be input via an image scanner, and the two-dimensional vertices, edges and line type are segmented by a low-level image processing stage. Several methods for achieving this have been reported in [22]; hence for our present work, we assume such output which describes the geometry of the three views is available. We assume that center lines of rotational surfaces or circles (ellipses) are included in the drawing. Center lines are used to indicate the centers of circles (ellipses) or the axis of a circular or rotational parts, such as cylinders. Fig. 4 shows the center lines of a cylinder.

In this step we process the input data and remove certain redundancies. It performs the following operations:

1. Separate and identify all views
2. Check the data for validity and reduce it to canonical form with distinct edges and vertices and edges intersecting only in endpoints [10]. If two conics (regard straight lines as a special form of conics) intersect internally, calculate the intersection points between the conics and partition them at the intersection points
3. Determine the extreme points of conic sections, which are the points that have the maximum or minimum value in one of the co-ordinate axes
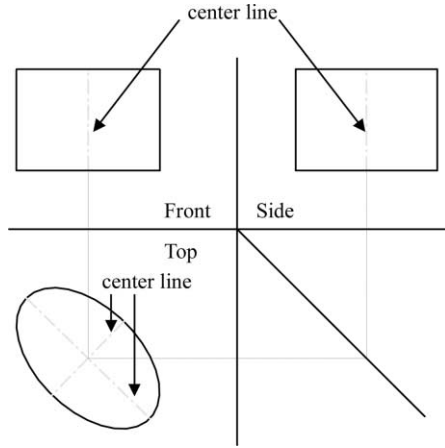4. Remove redundant curves and combine collinear curves

Fig. 4. Center line.

Fig. 5. Separation of three views.

Fig. 6. The match of conic edges in three orthographic views.

For scanned paper drawings, the 2-D segmented geometry should be corrected before reconstruction begins. Such operations are possible if we associate the dimensions indicated in the drawing with the corresponding geometric primitives and analyze the geometrical and topological relations in each 2-D view [23,24]. After these corrections, we separate and identify each 2-D view. According to the characteristics of three orthographic views, the union of the projections of all the curves in three views onto a co-ordinate axis produces two intervals (see *AB* and *CD* in Fig. 5). Therefore, the projections of the three views onto the two co-ordinate axes produce four intervals, two of which are on the *x*-axis (*AB* and *CD*), and the other two on the *y*-axis (*EF* and *GH*). Next we draw a vertical line l_y between the two intervals *AB* and *CD* and a horizontal line l_x between the other two intervals *EF* and *GH*. The view plane is then divided into four parts by l_x and l_y, which we denote as P1, P2, P3, and P4. From the convention of engineering drawings, the front view is in P1, side view in P2, and top view in P3. We then separate the three views by collecting the curves that belong to the same part.

3.2. Generation of wire-frame

In this step all the candidate 3-D vertices and edges that constitute the wire-frame are constructed. For brevity, we write 3-D candidate vertices, edges and faces as c-vertices, c-edges, c-faces, respectively.

We adopt the method described in Wesley [10] and Liu [16] for reconstructing the c-vertices. A c-vertex is created from three 2-D vertices, each appearing in one views. Let $N_f = (N_f(x), N_f(z))$, $N_t = (N_t(x), N_t(y))$, and $N_s = (N_s(y), N_s(z))$ be 2-D vertices in the front, top, and side view respectively. If

$$\left|N_f(x) - N_t(x)\right| < \epsilon, \left|N_t(y) - N_s(y)\right| < \epsilon, \ldots \left|N_f(z) - N_s(z)\right| < \epsilon \quad (3)$$

where ϵ is a tolerance that allows for inexact match, then we know that N_f, N_t, and N_s are the projections of a 3-D vertex
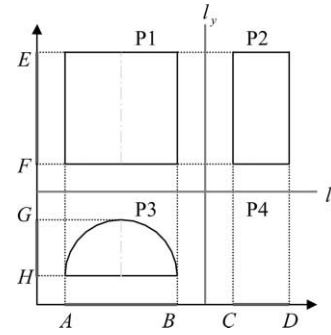
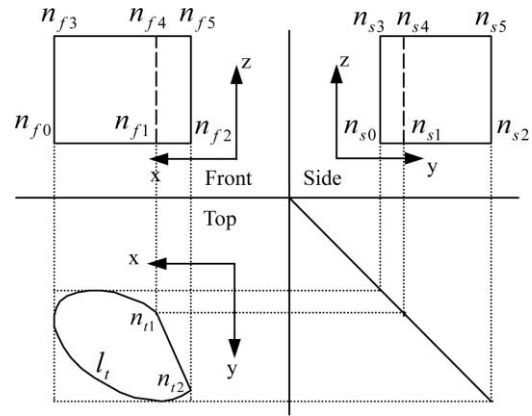in the different views. A c-vertex $(N_f(x), N_s(y), N_s(z))$ is thus generated.

Next, we propose a new method for generating c-edges. We begin by finding their matched components in the three orthographic views. We note that for a 2-D conic in one view, its matched components in the other two views can not be obtained by matching the endpoints of the conic, but can be obtained by matching the extreme points of the conics. We call this method *the extreme point method*.

The extreme point method is illustrated in Fig. 6. Let n_{fi} denote a 2-D vertex in the front view. Similar notations are introduced for the other two views. The 2-D conic edge l_t with endpoints n_{t1} and n_{t2} in the top view corresponds to the 2-D edges $n_{f0}n_{f2}$ and $n_{f3}n_{f5}$ whose two ends match the extreme points of l_t, and not the endpoints of l_t which match the edge. Similarly l_t corresponds to the 2-D edges $n_{s0}n_{s2}$ and $n_{s3}n_{s5}$ in the side view.

The next step is to find the c-edges from the 2-D matched components. Since examining all possible pairs of vertices would be inefficient, we use the same strategy as in Shin [13] to reduce the number of examinations. That is, for each 2-D conic (line) in the three views, only the pairs of c-vertices which are generated from the two endpoints of the 2-D conic edge are checked to determine whether there are matched components in the other two views. This greatly reduces the searching space, however, for a given 2-D conic, there could be more than one 2-D matched

components in one view. For example, in Fig. 7, f_A, t_A, s_A and f_B, t_B, s_B are two sets of respectively matched points in the three views. Simply using any matched component in a view to generate the c-edges may result in invalid edges. For example, for the 2-D edge $t_A t_B$ in the top view, the matched components in the front view are two edges $fl1$, $fl2$ which are between f_A and f_B, and in the view are two edges $sl1$, $sl2$ which are between s_A and s_B. Therefore, we need to find other points to define a unique 3-D conic edge [15]. Since during the processing stage, all the 2-D curves are partitioned at their intersection points, we can now assume that two curves in each view share at most two points (i.e., endpoints). Therefore, we can determine a unique 3-D edge from three distinct 3-D points and two conjugate diameters of the projection of the 3-D edge in a view. In Fig. 7, t_C in the top view is chosen to uniquely determine two matched points of $t_A t_B$ in the other two views.

The mathematical discussion in Section 2 set the background for developing an algorithm for automatically generating the c-edges from matched 2-D edges in three views. We call our method the *conjugate diameter method*. It consists of the following steps.

1. (Match). For a 2-D conic in a view, find the 2-D matched components in the other two orthographic views by the three-point method.
2. (Generate conjugate diameters in space). For each 2-D conic in one view, search for the matched line segments of its major and minor axis in the other two views, then construct the conjugate diameter (chord) of the space conic using the straight line edge generation method described by Wesley [10].
3. (Generate conic edge in space). By Theorem 3 and Corollary 2, determine uniquely a conic edge in space from its two conjugate diameters.

3.3. Generation of 3-D candidate faces

The c-vertices and c-edges generated in the previous stage constitute a wire-frame model of the object, but it does not contain any face information and may contain some invalid components. Since boundary representations are generally more useful than wire-frames in geometric modeling applications [25], we should generate the face information of the object from the reconstructed wire-frame. We shall use the term surface to refer to an unbounded face, and the term face to refer to the region of a surface that is bounded by an edge loop. We begin this step by creating all the surfaces that contain the solid's faces based on the fact that a surface can be determined from two edges which share a common c-vertex on the surface. The types of the two edges and their relationships uniquely determine the type and features of the surface [12]. For example, two coplanar edges sharing a c-vertex determine a planar surface. A line (conic) and a non-coplanar conic
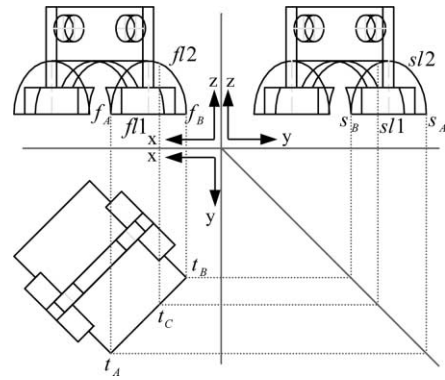

Fig. 7. Determine a unique candidate edge.

constitute a curved surface. If we generate all possible surfaces at this step, it will cause a steep increase in the processing time for complex objects. We use the following property to improve the performance.

Property 1. Each face of a 3-D object corresponds to at least a closed area in one of the views.

To begin with, we find all closed areas in one view. Next, for each 2-D closed area identified, we choose two 2-D edges which share a common 2-D vertex. Furthermore, we select two c-edges that are generated from these two 2-D edges (for each 2-D edge, the indices of the c-edges that are generated from it are stored as an attribute of that edge) and make sure that the selected two c-edges also share a common c-vertex. The types of these two edges and their relationships will determine the type of the surface to be generated. Finally, we gather all the c-edges that belong to this surface by only testing those c-edges that are generated from the 2-D edges in the closed area. After this stage, we have obtained all the support surfaces for the object and all the c-edges are contained in these surfaces.

In each surface generated in the previous stage, we must trace all the edge loops that constitute faces of the solid object. We propose a method called *maximum turning angle* (MTA) to automatically trace the three-dimensional edge loops. Its basic idea is to first choose a convex c-vertex and a c-edge that is connected to it, and then search the next edge using adjacency relationships. As shown in Fig. 8, let e_1 be the current edge being examined, and the direction v_i to v_j to be the traversing direction. When there is more than one edge that satisfies the surface equation at the vertex v_j,
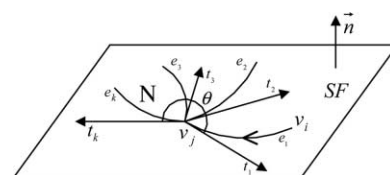

Fig. 8. Determining the edge to be contained in the current edge loop

we choose the edge e_k which has the maximum turning angle to e_1. The turning angle is the angle by which the tangent vector of a curved edge has to be rotated at the common vertex while going from the tangent vector of the present curved edge to the tangent vector of a connecting curved edge. The turning angle is calculated around the hypothetical outer normal of the face at the vertex v_j. Counterclockwise rotations are assumed positive. Arrange the c-edges incidented to v_j in descending order of the turning angle with respect to the selected c-edge e_1 and denoted them as an ordered list $E(v_j)$. For example, in Fig. 8, $E(v_j) = \{e_1, e_k, e_{k-1}, \ldots, e_2, e_1\}$. The next adjacent edge is the edge in $E(v_j)$ that has the maximum turning angle to the current examined edge. In Fig. 8, e_k has the maximum turning angle to e_1, e_{k-1} has the maximum turning angle to e_k, and so on. This is useful for determining the next c-edge in the process of searching an edge loop.

Next, we define what we mean by a point is to the left of an edge, which is useful for choosing the first edge of an edge loop on a face.

Definition 8. (left). A directed line is determined by an ordered pair of points AB (see Fig. 9). A point C is to the left of AB if and only if the triangle ABC has a positive area. Then, the triple (A, B, C) forms a counterclockwise circuit. If both the endpoints of a line segment CD are to the left of or on the line AB, we say CD is to the left of AB.

The MTA method efficiently finds the edge loops in the wire-frame by avoiding an exhaustive search. It consists of the following six steps.

Step 1. Let SF be a surface in a solid object. We denote the set of edges on the surface SF as $E(SF)$. Choose a convex vertex v_i (usually the vertex that has the minimum x-co-ordinate) of SF as the starting vertex, and let $v_s = v_i$.

Step 2. Determine all the edges incident to each vertex v, and denote it as $AE(v)$. If $|AE(v)| > 2$, determine the ordered list $E(v)$ for the vertex v.

Step 3. If $|E(SF)| > 0$, select the starting edge to be an edge $e_s(v_s, v_j)$ in $AE(v_s)$ that has not been chosen or has been chosen only once. If the edge has not been selected, we must ensure that all the other edges are to its left, so that a counterclockwise orientation can be assumed, which avoids producing edge loops with other edges in $E(SF)$ subdividing them. If e_s has been selected once, assign $v_j = v_s$, $v_s = v_j$. Let $\overrightarrow{v_s v_j}$ be the current direction of travel, go to step 4.

Step 4. If $|AE(v_j)| = 2$, choose the other edge that is adjacent to v_j; otherwise, choose the edge with the maximum turning angle in $E(v_j)$ as the next tracing edge. Let the chosen edge be $e_l(v_j, v_l)$.

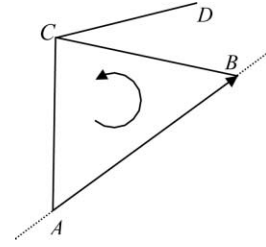**Step 5.** If $v_l \neq v_s$, let $v_j = v_l$ and go to step 4; otherwise,

Fig. 9. C is to left of AB if the triangle ABC has a positive area.

construct an edge loop, and delete from $E(SF)$ those edges that satisfy the deleting conditions given below:

1. it is the starting edge of an edge loop;
2. one of its endpoints has degree 1;
3. it has been chosen twice. A c-edge may be regard as if it is shared by two distinct faces on the same surface [13] such as c_1 in Fig. 11(a) because of the loss of information during the projection. For this reason, a c-edge may at most be chosen twice to the opposite directions.

In the process of tracing the edge loop, if $|E(v)| = 2$ for all the vertex v in the edge loop, choose an unselected convex vertex of SF and record it as v_{ns}; otherwise record the first encountered vertex whose degree is greater than 2 as v_{ns}. The vertex v_{ns} will be the entry for tracing the next edge loop.

Step 6. If $|E(SF)| > 0$, then treat v_{ns} as v_s and go to step 3. Otherwise stop the algorithm.

Remark:. When arranging $AE(v_i)$ in descending order of the turning angle with respect to an arbitrary selected c-edge e_1, we avoid arc sine function by comparing their dot products with the co-ordinate axes, which is illustrated as follows. Let the tangent vectors of the edges adjacent to v_i be $t_1, t_2, \ldots t_k$. Let t_1 be the x-axis and the outer normal u of the surface at v_i be the z-axis, then the y-axis is $u \times t_1$. Let n_x and n_y be the unit vectors in the direction the x-axis and the y-axis respectively, the angle between e_1 and e_i be θ_i, we then have

$$\begin{cases} n_x \cdot e_i = \cos\theta_i \\ n_y \cdot e_i = \sin\theta_i \end{cases} (i = 2, \ldots, k) \qquad (4)$$

by comparing $\cos\theta_i$ and $\sin\theta_i$ $i = 2, \ldots k$ we can determine the order of these angles.

Fig. 10 shows the edge loop generation process based on the MTA method. The dashed lines in the figure are edges that are deleted in the step; three edge loops are traced in this example, which are shown at the bottom of the figure.
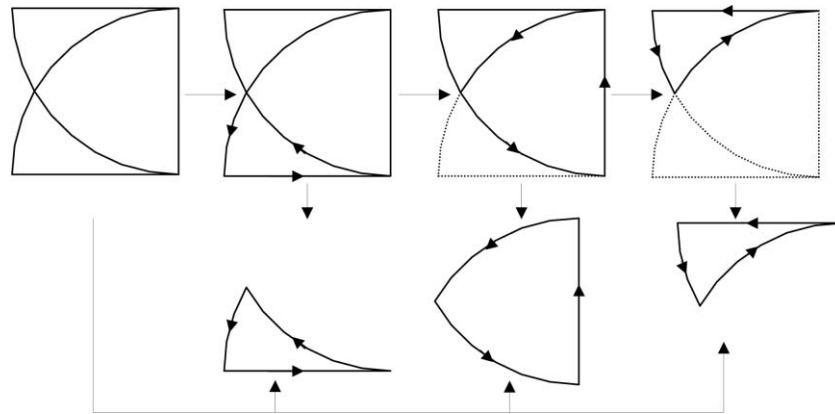
Fig. 10. Top: the process for edge loop generation. Bottom: three edge loops constructed from this surface.

3.4. Searching for solids

The algorithm to be discussed works for boundary representations of solids with curved surface. More rigorously, we assume that our reconstructed objects are manifold solids, i.e., they satisfy the following criteria [26,27]:

1. Every edge in a manifold solid belongs to exactly two faces and the orientation of the edge is inverted by each face;
2. Faces may not intersect each other except at common edges or vertices.

Since the projection process may introduce ambiguity into the reconstruction problem, the reconstruction process usually generates many more 3-D vertices and edges that are not in the original objects. Therefore, we must detect and delete pseudo elements in the reconstructed wire-frame.

Definition 9. (non-manifold edge). If an edge in the wire-frame is shared by only one c-face, or more than two c-faces, then the edge is called a non-manifold edge.

We adopt a heuristic approach in this step to reduce the graph search space and the occurrences of backtracking. Initially we use depth information to remove some false c-edges and c-faces. The depth information that is deduced from the type of line, namely dashed and solid, in the drawing is fully utilized at this stage in order to reduce the searching space. For example, if the projection of a c-edge on the front view is a dashed curve, there should be at least one c-face that obstructs the c-edge when viewing from the negative direction of the y-axis. If there is no such c-face, then this c-edge must be removed. On the other hand, we could conclude that the c-faces that obstruct the visible c-edges are pathological and should be deleted. The remaining pseudo faces in the wire-frame are then completely removed by a divide-and-conquer approach based on the Moebius rule, the definition of manifold, and the characteristics of engineering drawings. The Moebius rule says that each edge

in a manifold solid belongs to two faces and the orientation of the edge is inverted by each face [12]. Therefore, the divide-and-conquer algorithm serves to generate all possible objects by detecting and removing non-manifold edges in the wire-frame. This algorithm is implemented by traversing the whole graph of possible face configurations, and backtracking when ever necessary, to find all possible solutions, while resolving ambiguities. The edge count of a 2-D curve in a view represents how many c-edges are projected onto that 2-D curve. While deleting the non-manifold edges, edge counts are used to dynamically monitor the projections of the wire-frame to determine whether or not it is identical with the input engineering drawing [12].

Fig. 11 illustrates the entire reconstruction. A three-view engineering drawing and the wire frame reconstructed from it is shown in Fig. 11(a). Fig. 11(b) shows the edges in the wire frame. The faces traced in this wire frame are given in Fig. 11(c) and (d)–(l). illustrate the steps for finding the 3-D object from the wire frame. To begin with, we use the depth information in the engineering drawing to check the validity of the c-edges and c-faces in the wire frame. In the wire frame, the projections of the c-edges e_1, e_{12}, and e_{23} in the top view are the dashed conic c_1 shown in Fig. 11(a). The c-edge e_1 is chosen to be the starting edge as in Fig. 11(d), it is invalid since there is no c-face that occludes it. Therefore the c-faces f_1, f_3, and f_4 which include e_1 are invalid. Since the faces f_1 and f_3 which are adjacent to the invalid edge e_1 are coplanar, we combine those two faces to a larger face f_{13} and assume this face is valid. The c-edge e_{11} is adjacent exactly to two faces f_5 and f_{14}, hence the c-faces f_5 and f_{14} and the c-edge e_{11} are valid by the Moebius rule. Moreover, if the invalid face f_4 is deleted, the c-edge e_5 is only adjacent to f_2 and e_{10} is adjacent to f_{16}. According to the definition of manifold, e_5, e_{10}, f_2, and f_{16} should be deleted as shown in Fig. 11(e). Then non-manifold c-edge e_{18} becomes the starting edge for the next decision process as shown in Fig. 11(f). The face f_{14} is valid; this lead to two possible cases: assume f_{14}, f_8 to be valid and f_{15} to be pseudo as in Fig. 11(g); assume f_{14}, f_{15} to be valid and f_8 to be pseudo as in Fig. 11(j). For the first case, f_{15} being valid
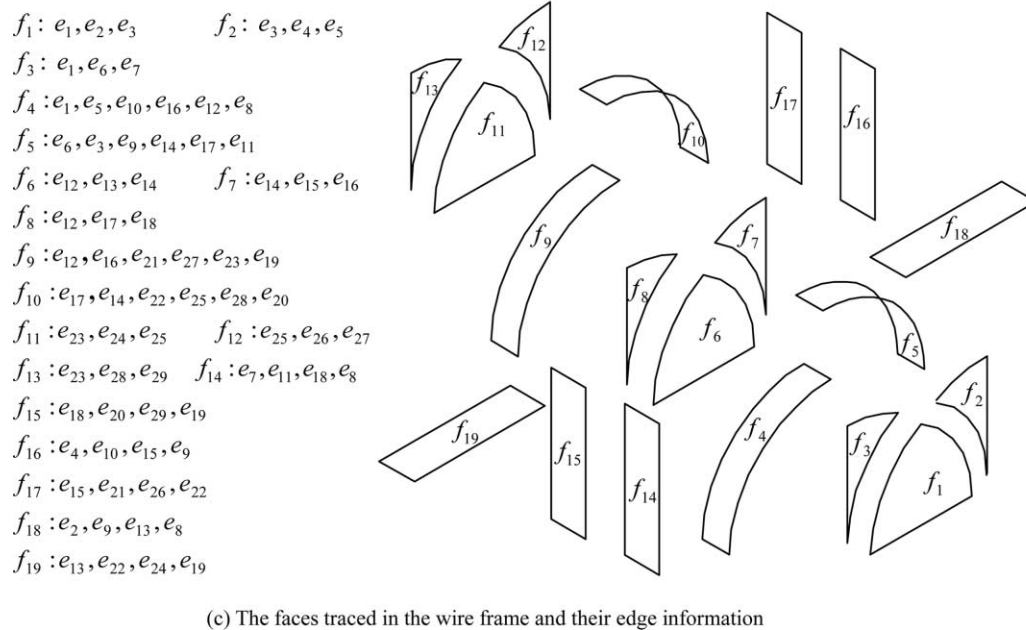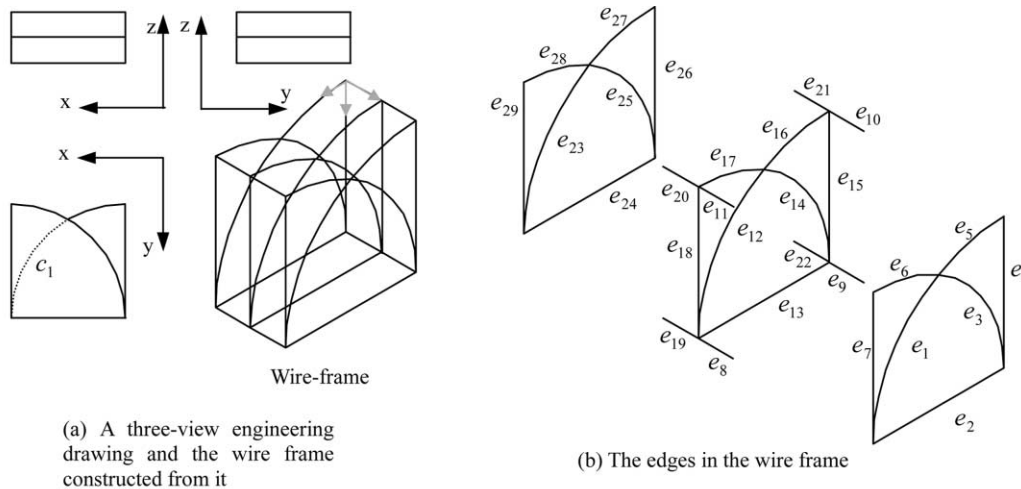
(a) A three-view engineering drawing and the wire frame constructed from it

(b) The edges in the wire frame

$f_1: e_1, e_2, e_3$ $f_2: e_3, e_4, e_5$
$f_3: e_1, e_6, e_7$
$f_4: e_1, e_5, e_{10}, e_{16}, e_{12}, e_8$
$f_5: e_6, e_3, e_9, e_{14}, e_{17}, e_{11}$
$f_6: e_{12}, e_{13}, e_{14}$ $f_7: e_{14}, e_{15}, e_{16}$
$f_8: e_{12}, e_{17}, e_{18}$
$f_9: e_{12}, e_{16}, e_{21}, e_{27}, e_{23}, e_{19}$
$f_{10}: e_{17}, e_{14}, e_{22}, e_{25}, e_{28}, e_{20}$
$f_{11}: e_{23}, e_{24}, e_{25}$ $f_{12}: e_{25}, e_{26}, e_{27}$
$f_{13}: e_{23}, e_{28}, e_{29}$ $f_{14}: e_7, e_{11}, e_{18}, e_8$
$f_{15}: e_{18}, e_{20}, e_{29}, e_{19}$
$f_{16}: e_4, e_{10}, e_{15}, e_9$
$f_{17}: e_{15}, e_{21}, e_{26}, e_{22}$
$f_{18}: e_2, e_9, e_{13}, e_8$
$f_{19}: e_{13}, e_{22}, e_{24}, e_{19}$

(c) The faces traced in the wire frame and their edge information

Fig. 11. (*caption on following page*)

causes the faces f_{10} and f_{13} to be pseudo as in Fig. 11(h). Each of the c-edges e_{14} and e_{25} is adjacent exactly to two coplanar faces, so they are deleted from the wire frame. Thus a solution is found in Fig. 11(i). Similarly, for the second case, we could detect that the c-edges e_{12}–e_{18}, e_{21}, e_{23}, e_{26}, e_{27}, and the c-faces f_6, f_7, f_8, f_9, f_{17} are pseudo as shown in Fig. 11(j) and (k). Thus another solution is found as shown in Fig. 11(l). This solution is invalid since it is not identical with the input three-view engineering drawing.

4. Implementation

Based on the algorithm described above, we have developed a 3-D reconstruction program on a personal computer platform. A number of examples have been tested. The implementation and examples are restricted to three orthographic views and objects with quadric and toroidal surfaces. Several examples are given to demonstrate various cases that could be handled by our method.

Fig. 12 shows a three-view engineering drawing that contains straight lines, circular arcs, elliptical arcs, and hyperbolas; our algorithm generates a unique solution. Fig. 13 shows a mechanical part with planar and cylindrical faces, the semi and quadrant cylindrical faces meet the planar faces tangentially. Fig. 14 shows an example with cylinders (including the semi cylinder) whose axes are not parallel to any of the three co-ordinate axes.

Figs. 15 and 16 show the reconstruction of some real mechanical parts. The axes of the cylindrical surfaces that

(d)-(l) Steps for determining the state of the edges and faces
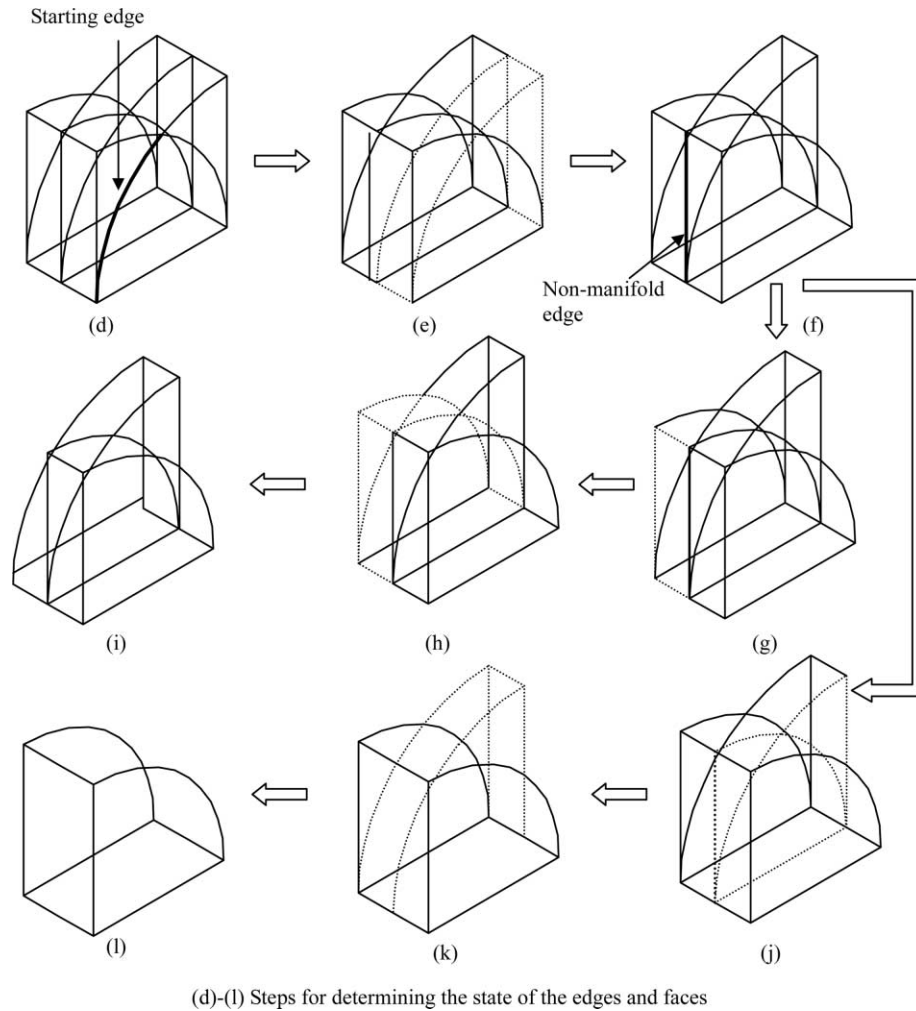
Fig. 11. Finding the solid(s) in the wire frame. (a) A three-view engineering drawing and the wire frame constructed from it, (b) The edges in the wire frame, (c) The faces traced in the wire frame and their edge information, (d)–(l) Steps for determining the state of the edges and faces.

are perpendicular to the slant plane in Fig. 16 are not parallel to any of the three co-ordinate axes. The example in Fig. 17 illustrates the use of depth information. Without depth information, there are two possible solids. However, by referring to the dashed lines in the top view, a unique solution is generated, i.e., solution 1 is the final solution.

The next example is intended to demonstrate the distinc-

tion between our method and Shin's method. The result is shown in Fig. 18. It follows from the visual comparison that Shin's method will result in approximation errors in the case of curved surfaces. If curved shapes are approximated by more polygons, the errors may be reduced. However, this tends to increase the computation. Compared to Shin's work, our approach avoids the drawbacks of polyhedron

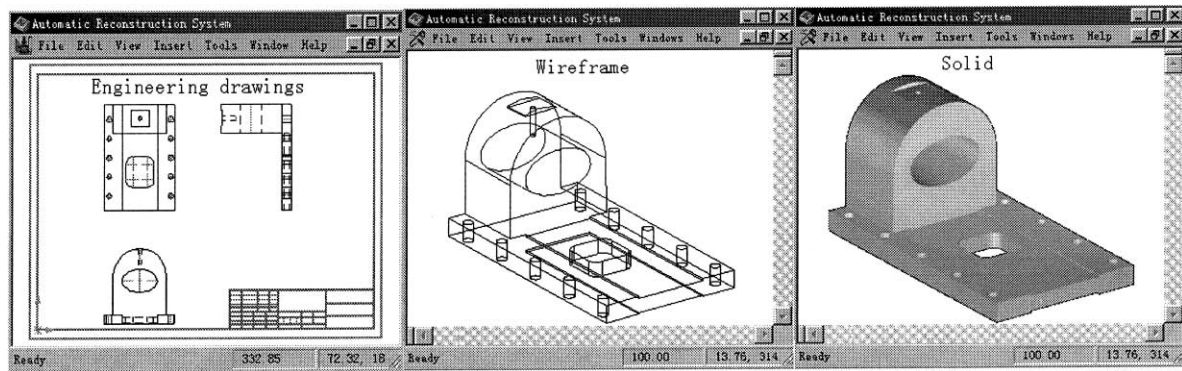Fig. 12. Engineering drawings and reconstructed model for object 1.

Fig. 13. Engineering drawings and reconstructed model for object 2.

approximation by utilizing the characteristics of engineering drawings and the geometric properties of conic curves. Moreover, our method covers a larger category of solid objects.

Finally, we compare our method with other methods in terms of the input primitives allowed and the restrictions on the axes of conics and curved surfaces. The results are shown in Table 1.
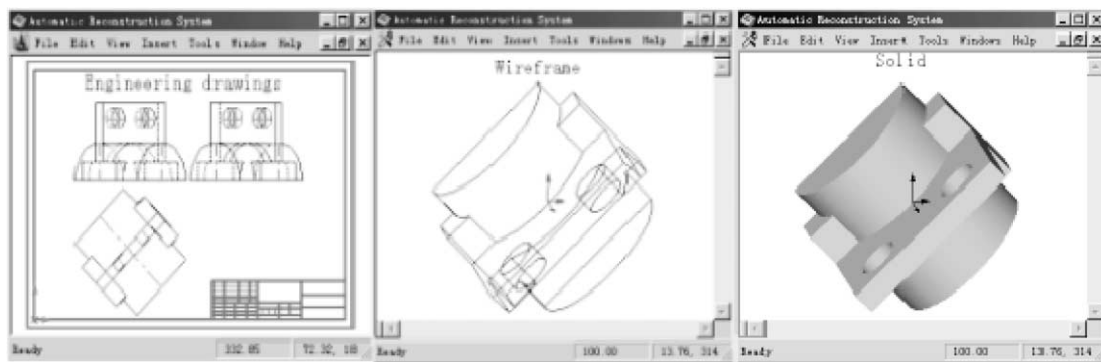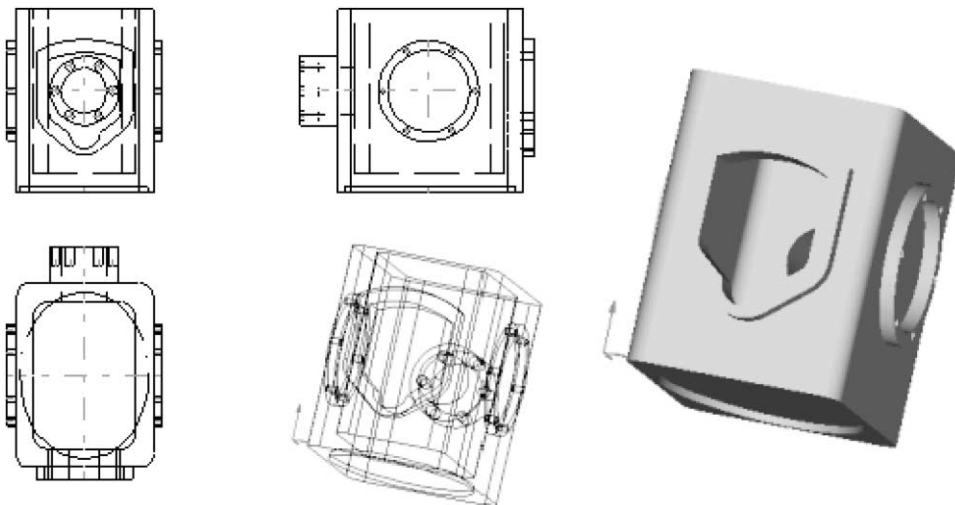

Fig. 14. Engineering drawings and reconstructed model for object 3.

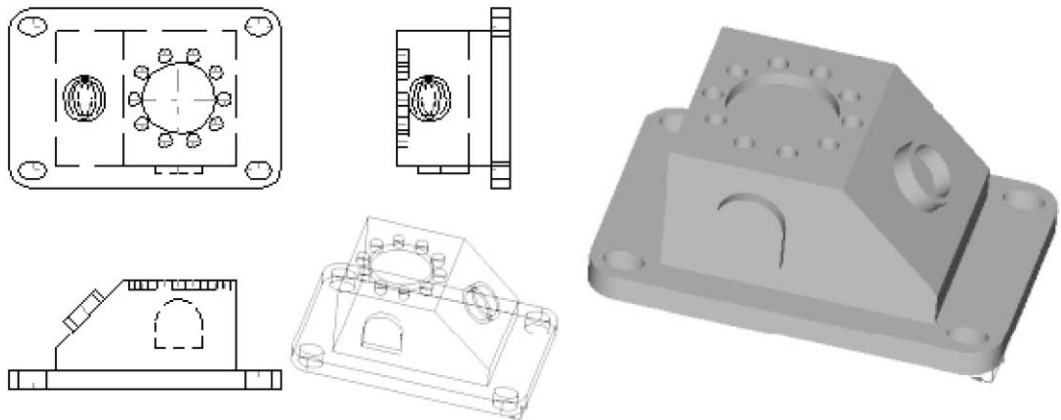Fig. 15. Engineering drawings and reconstructed model for object 4.

Fig. 16. Engineering drawings and reconstructed model for object 5.
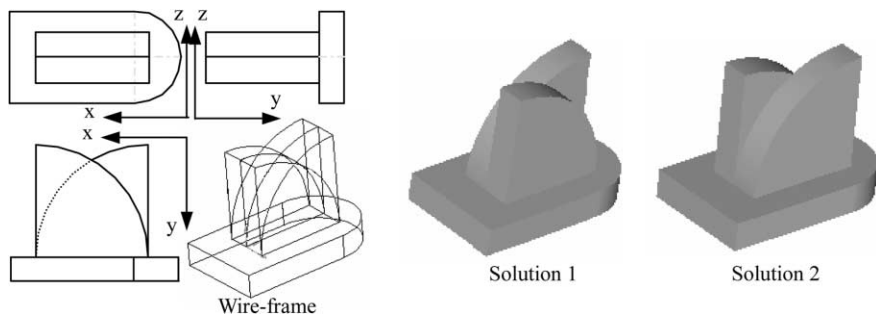

Fig. 17. Three pairs of circular arcs intersect in the reconstructed wire-frame. Before using the depth information, there are two possible solutions. Referring to the dashed line in the top view, a unique solution is acquired.
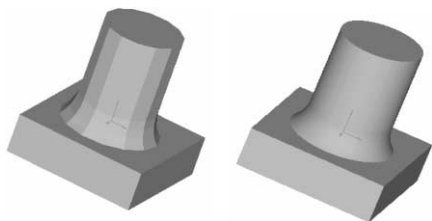

Fig. 18. A comparison illustrating the exactness of our approach. On the left is a solid object reconstructed by Shin's method, while on the right is our results.

5. Conclusion and future work

We present the first solution to the problem of reconstructing solids from drawings that places no restrictions on the axes of curved surfaces and conics. The main features of the algorithm are that it extends the range of objects that can be reconstructed and improves the speed of the reconstruction. To support a wide scope of general 3-D objects, we propose a matching principle and a novel methodology for generating the conic edges. Furthermore, we introduce some decision and heuristic rules to generate the final 3-D objects.

The development of a prototype reconstruction system is reported in this paper. In the prototype system, wire-frame models, surface models, and solid models are constructed from orthographic views in that order. The conjugate diameter method is introduced to increase the efficiency of edge generation. In addition, the depth information that is deduced from the type of line, namely dashed and solid, in the drawing is fully utilized in order to reduce the searching space. Finally, we handle the problem of ambiguities in

Table 1
Comparison of the input data and restrictions on the axis

Methods	Input data					Restrictions
	Line	Circle	Ellipse	Parabolas	Hyperbolas	
Shin	Y	Y	N	N	N	Parallel to one co-ordinate axis
Oh	Y	Y	N	N	N	Parallel to one co-ordinate axis
Our method	Y	Y	Y	Y	Y	No restrictions

wire-frame representations of engineering drawings by the definition of manifold and Moebius rule. This algorithm is attractive when exhaustive examination of all combinations is not practical.

We regard the work presented as initial, there are improvements to be made as well as many directions to pursue. The future work will be focused on extending the range of objects to be reconstructed. Geometric and topological information from each component should be fully exploited to improve the efficiency of the reconstruction. Better use of adjacency information may lead to more efficient algorithms. In addition, we should make full use of the annotation in engineering drawings since it is a formal, standard language which expresses a designer's design purpose.

Acknowledgements

References

[1] Idesawa M. A system to generate a solid figure from three view. Bulletin of the JSME 1973;16(92):216–25.

[2] Nagendra IV, Gujar UG. 3-D objects from 2-D orthographic views-a survey. Computer & Graphics 1988;12(1):111–4.

[3] Wang W, Grinstein GG. A survey of 3-D solid reconstruction from 2-D projection line drawings. Computer Graphics Forum 1993;12(2):137–58.

[4] Zhao H, Kim YS. Geometric operations for visual reasoning of a solid from orthographic projections. Advances in Engineering Software 1999;30(7):445–50.

[5] Chen Z, Perng DB. Automatic reconstruction of 3-D solid objects from 2-D orthographic views. Pattern Recognition 1988;21(5):439–49.

[6] Meeran S, Pratt MJ. Automated feature recognition from 2-D drawings. Computer-Aided Design 1993;25(1):7–17.

[7] Hiroshi Masuda, Masayuki Numao. Research—A cell-based approach for generating solid objects from orthographic projections. Computer-Aided Design 1997;29(3):177–87.

[8] Shum SP, Lau WS, Yuen MF, Yu KM. Solid reconstruction from orthographic opaque views using incremental extrusion. Computer & Graphics 1997;29(6):787–800.

[9] Markowsky G, Wesley MA. Fleshing out wire frames. IBM J. Res. Develop. 1980;24(5):582–97.

[10] Wesley M, Markowsky G. Fleshing out projection. IBM J. Res.Develop. 1981;25(6):934–54.

[11] Sakurai H, Gossard DC. Solid model input through orthographic views. SIGGRAPH 83 Proceedings 1983:243–52.

[12] Lequette R. Automatic construction of curvilinear solids from wire-frame views. Computer Aided Design 1988;20(4):171–9.

[13] Shin B, Shin Y. Fast 3-D solid model reconstruction from orthographic views. Computer Aided Design 1998;30(1):63–76.

[14] Kuo MH. A systematic approach towards reconstructing 3-D curved models from multiple 2-D views. In: Tombre K, Chhabra A, editors. Graphics Recognition—Algorithms and Systems. Lecture Notes in Computer Science, Vol. 1389. Springer, 1998. p. 265–79.

[15] Kuo MH. Reconstruction of quadric surface solids from three-view engineering drawings. Computer Aided Design 1998;30(7):517–27.

[16] Liu SX, Hu SM, Sun JG. Reconstruction of curvilinear objects from three orthographic views. AD/CG'99 1999:1040–4.

[17] Oh BS, Kim CH. Systematic reconstruction of 3-D curvilinear objects from two-view drawings. Computers & Graphics 1999;23(3):343–52.

[18] Tanaka M, Iwama K, Hosoda A, Watanabe T. Decomposition of a 2-D assembly drawing into 3-D part drawings. Computer-Aided Design 1998;30(1):37–46.

[19] Bix Robert. Conics and cubics: a concrete introduction to algebraic curves. New York: Springer-Verlag, 1998.

[20] Farin G. Curves and surfaces, computer aided geometric design. 4rd edn.. New York: Academic Press, 1997.

[21] Wilson PR. Conic representations for shape description. IEEE Computer Graphics and Applications 1987;7(4):23–30.

[22] Dori D, Tombre K. From engineering drawings to 3-D CAD models: are we ready now? Computer Aided Design 1995;27(4):243–54.

[23] Collin Coll S. Syntactical analysis of technical drawing dimensions. Proceedings of international workshop on structural and syntactic pattern recognition. Bern: World Scientific, 1992. p. 280–9.

[24] Dori D, syntactic/geometric A. approach to recognition of dimensions in engineering drawings. Computer Vision, Graphics and Image Processing 1989;47:271–91.

[25] Counter S. Mark, Brewer John A. Automated conversion of curvilinear wire-frame models to surface boundary models: a topological approach. In SIGGRAPH 86 proceedings, 1986, pp. 171-178.

[26] Kumar V, Burns D, Dutta D, Hoffmann C. A framework for object modeling. Computer Aided Design 1999;31(9):541–56.

[27] Mantyla M. Boolean operation of 2-manifolds through vertex neighborhood classification. ACM Transactions on Graphics 1986;5(1):1–29.

Shi-Xia Liu is a Ph.D. student in the Department of Computer Science and Technology, Tsinghua University, Beijing, P. R. China. She received a BSC in Computing Mathematics from Harbin Institute of Technology in 1996, a MSC degree in Computational Geometry and Computer Graphics from Harbin Institute of Technology in 1998. Her research interests are computer aided design and computer graphics.

Yu-Jian Chen is a professor in Department of Computer Science and Technology, Tsinghua University. Her research interests are in computer aided geometric design, and computer graphics.

Shi-Min Hu is an associate professor in Department of Computer Science and Technology, Tsinghua University. He received a BSC in Mathematics from Jilin University in 1990, a MSC and a Ph.D. degree in Computational Geometry and Computer Graphics both from Zhejiang University in 1993 and 1996. His research interests are in computer aided geometric design, computer graphics, and content-based image indexing.

Jia-Guang Sun is a professor in Department of Computer Science and Technology, Tsinghua University. He is also director of National CAD Engineering Center at Tsinghua University, and Academician of Chinese Academy of Engineering. His research interests are in computer aided geometric design, computer graphics, and product data management.