Two Accelerating Techniques for 3D Reconstruction

LIU Shixia (刘世霞), HU Shimin (胡事民) and SUN Jiaguang (孙家广)

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

E-mail: liusx@ncc.cs.tsinghua.edu.cn; {shimin,sunjg}@tsinghua.edu.cn

Received March 15, 2001; revised August 7, 2001.

Abstract Automatic reconstruction of 3D objects from 2D orthographic views has been a major research issue in CAD/CAM. In this paper, two accelerating techniques to improve the efficiency of reconstruction are presented. First, some pseudo elements are removed by depth and topology information as soon as the wire-frame is constructed, which reduces the searching space. Second, the proposed algorithm does not establish all possible surfaces in the process of generating 3D faces. The surfaces and edge loops are generated by using the relationship between the boundaries of 3D faces and their projections. This avoids the growth in combinational complexity of previous methods that have to check all possible pairs of 3D candidate edges.

Keywords engineering drawing, wire-frame, reconstruction

1 Introduction

Reconstructing 3D objects from orthographic views has been studied for the past two decades[1,2]. Algorithms attempting to solve automatic reconstruction can be divided into CSG oriented and B-rep oriented approaches according to how they construct and represent the final solid objects.

The CSG oriented approach assumes that each 3D object can be built from certain primitives in a hierarchical manner. It selects a 2D loop as a base and generates the 3D primitives by matching their corresponding 2D patterns in each view, then assembles the primitives to form the final object. Reconstruction work proposed by Chen *et al.*[3], Meeran *et al.*[4], Masuda *et al.*[5] and Shum *et al.*[6,7] was based on this approach. However, this method could only handle objects of uniform thickness or rotational objects, which restricts their domain of applicability.

The B-rep oriented approach is a bottom-up approach, which uses the constraint propagation technique. In this method, a wire-frame which is composed of 3D vertices and 3D edges is first recognized from the input orthographic views. Then, candidate faces are found from all these 3D edges. Finally, pseudo elements are deleted by certain criteria and all true faces are assembled to form the final solid object.

Idesawa[8] proposed an algorithm which could only construct very limited polyhedral objects from three views. Markowsky and Wesley[9] provided a comprehensive algorithm for searching all possible solutions of polyhedral objects from the given wire-frame, which was then extended to reconstruct polyhedral objects from three orthographic views[10]. Sakurai *et al.*[11] extended the approach of Wesley and Markowsky to include planar, cylindrical, conical, spherical and toroidal surfaces, whose projections are straight lines and circles (arcs). Kuo[12] proposed a five-point method to generate 3D conic edges and used a decision-chaining method to detect and delete the pseudo elements. Oh *et al.*[13] presented an algorithm for reconstructing curvilinear 3D objects from two-view drawings. They first constructed the partial objects, and then obtained the complete objects from the partially constructed objects by adding perpendicular faces with geometric validity. In a previous paper, we[14] proposed a geometric method to reconstruct 3D objects from three-view engineering drawings, which placed no restrictions on the axis of symmetric surfaces and the center lines of conics.

This paper presents two accelerating techniques to improve the efficiency of the B-rep oriented re-

This work is supported by the National Natural Science Foundation of China under grant No.69902004 and the NKBRSF of China under grant No.G1998030600.

construction algorithm. By removing some pseudo edges directly in the wire-frame and using the relationship between 3D elements and its projections, our algorithm reduces the searching space and the number of examinations.

2 Solid Model Reconstruction

In this section, we introduce the main procedure of 3D reconstruction. More details can be found in [12, 15].

We use a B-rep oriented approach to reconstruct 3D objects from three orthographic views. The proposed method constructs 3D candidate vertices, edges, and faces from the input engineering drawing in order. For brevity, we write 3D candidate vertices, edges and faces as c-vertices, c-edges, c-faces, respectively. The reconstruction algorithm can be divided into the following stages.

• Check input data

In this stage, we process the input data and remove certain redundancies to prepare for 3D reconstruction. The preprocessing performs the following operations. First, separate each view in the engineering drawing. Then, remove all duplicate 2D vertices and conics. Next, check each pair of conics, if they intersect internally, add the intersection point and divide each of the conics at the intersection point. Finally, combine the conics with the same equation.

• Reconstruct wire-frame

This stage constructs all c-vertices and c-edges that constitute the wire-frame model by certain correspondences among three orthographic views. We generate all c-vertices by selecting a 2D vertex from one view and then compare it with the vertices in the other two views. To reconstruct 3D conic edges, we use the *conjugate diameter method*[14] or the *matrix method*[15].

• Trace c-faces in the wire-frame

The main task of this step is to find the set of all possible edge loops in the wire-frame, from which subsets can be selected to form the boundary of the faces. First, we find the surfaces that contain the faces. Then, all the edge loops are constructed by using the relationship between the 3D face and its projections in the 2D views. This technique will be explained later.

• Search for 3D solids

Among all the c-faces generated in the previous step, some really lie on the boundary of a solid. These are real faces, and the others are called pseudo faces. Hence, we should find a group of faces that make up a real solid object. We use the same method as in our previous paper[15] to search all 3D objects within the reconstructed wire-frame. Initially, we use the depth information to remove some false faces. Then, the remaining pseudo faces in the wire-frame model are completely removed by a divide-and-conquer approach based on the definition of manifold.

3 Two Accelerating Techniques

3.1 Removing Pseudo Elements from Wire-Frame

The pseudo elements in the wire-frame will extend the searching space of the subsequent stages. To reduce the processing time later, we remove some redundant c-edges by the depth information in the engineering drawing and topology information in the 2-manifold object.

If a visible c-edge is generated from a dashed line in a view whose projection direction is the same as the viewing direction, then the c-edge is pseudo and should be eliminated from the wire-frame. This can be checked by the coordinate values along the projection direction. For example, if a c-edge has a dashed projection in the top view and its projection on the side view has a local maximum z-coordinate, then this edge is pseudo since it would be visible from the negative direction of z-axis, i.e., it corresponds to a solid line in the top view illustrated in Figs.1(a)–(c). The projection of c-edge $E(V_A, V_B)$ (see Fig.1(b)) onto the top view is a dashed line (see $C(t_A, t_B)$ in Fig.1(a)), while its projection onto the side view ($C(s_A, s_B)$ in Fig.1(a)) has a local maximum z-value (no other lines

occlude it from the negative direction of the z-axis). Therefore, $E(V_A, V_B)$ is a pseudo edge and should be removed from the wire-frame (see Fig.1(c)).

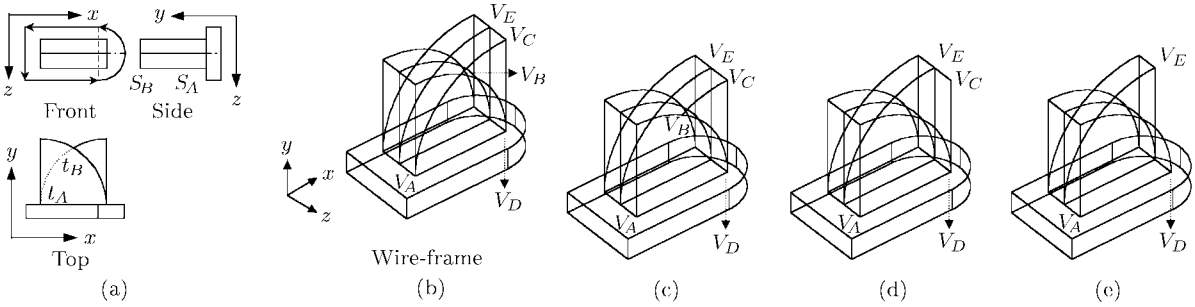

Fig.1. Remove pseudo c-edges by depth information.

Furthermore, we remove other pseudo elements by the topology information defined in 2-manifold objects. Several rules are given to eliminate the pseudo elements in the wire-frame, which are based on the topology information. For brevity, we introduce the definition of degree. The degree of a vertex v is defined as the number of the c-edges connected to it. We denote it as degree (v).

(1) If degree $(v) < 2$, remove vertex v and the c-edge connected to it.

(2) If degree $(v) = 2$ and the two c-edges connected to it are collinear, merge the two c-edges and remove vertex v.

(3) If degree $(v) = 2$ and the two c-edges connected to it are not collinear, remove vertex v and the c-edges.

(4) If degree $(v) = 3$ and two of these c-edges connected to it are collinear, merge the collinear c-edges, remove vertex v and the other c-edge.

(5) If degree $(v) \geq 3$ and all these c-edges connected to it are coplanar and not collinear, remove vertex v and all these c-edges.

Referring to Fig.1(c), the c-vertex V_B satisfies case (4), so we merge the two collinear c-edges and remove c-edge $E(V_B, V_C)$ as shown in Fig.1(d). Now, c-vertex V_C satisfies the condition described by case (3), so c-vertex V_C, c-edges $E(V_C, V_D)$ and $E(V_C, V_E)$ are deleted as in Fig.1(e).

3.2 Generating Surfaces and Faces

To generate all c-faces, the previous method (such as the method used by Kuo) finds all possible surfaces that contain the faces of the 3D object. These surfaces can be constructed from the information of two edges sharing a common vertex. The type of these two edges and the relationship between them determine the type and the features of the surface to be generated. For example, a planar surface can be built from two coplanar edges sharing a c-vertex. A line (conic) and a conic that are not coplanar with each other constitute a quadric or toroidal surface. Then, the previous method has to test all the c-edges in the wire-frame to find the c-edges which belong to the generated surface. This will cause a steep increased in the processing time for complex objects. Here, we reduce the searching time by using the relationship between the 3D face and its projections in the 2D views.

Oh *et al.*[13] used a special projection technique to simplify the process of searching the c-edges in a surface. They assumed that the edges on the original plane and the projection plane have a one-to-one relationship under such a special projection. The c-faces are then obtained by extracting the array of edges on the projection plane. Here, their method is improved and extended to handle three orthographic views. The new method has no restrictions on the projection technique.

Definition 1 (2D Loop). *A 2D loop in a view consists of an ordered cycle of 2D edges and 2D vertices.*

Definition 2 (Generalized Cylindrical Face). *A generalized cylindrical face is defined by moving an arbitrary 3D contour (may be not closed) along a 3D ray line.*

Definition 3 (Non-Degenerate Line). *A line that is not perpendicular to the projection plane is defined as a non-degenerate line to this projection plane.*

Property 1. *Under the parallel projection, if each line edge in a connected cycle of* $3D$ *edges is not perpendicular to the projection plane, then the projection of the closed cycle of* $3D$ *edges onto this view is a* $2D$ *loop. The correspondence between the* $3D$ *edges in the edge loop and the* $2D$ *edges in the* $2D$ *loop is one-to-one.*

Proof. The projection of the conic edge onto each of the three orthographic views is either a conic edge or a 2D line edge, so line edges are mainly considered in this property since their projections may degenerate into 2D points. The projection of a conic (line) edge onto a projection plane can be regarded as the intersection of the projection plane and a generalized cylindrical face. The contour of the generalized cylindrical face is the conic (line) edge. The 3D ray line is formed by moving a point on the conic (line) along the projection direction. Since the projection plane intersects a generalized cylindrical face which is not parallel to the projection plane only at a conic (line), therefore each 3D conic (non-degenerate line) corresponds to a 2D conic (line) in that projection. The extruding directions of these generalized cylindrical faces are the same for a given projection plane, so they do not intersect except at the common line. The common line is formed by moving the common 3D point along the projection direction. Its projection onto the projection plane is a 2D point. Thus, different 3D conic (non-degenerate line) edges correspond to different 2D conic (line) edges in the view. □

We use the following theorem to reduce the number of examinations, which states the correspondence between the 3D edge loop and the 2D loop in the orthographic views.

Theorem 1. *Each edge loop of a* $3D$ *object corresponds to at least a* $2D$ *loop in one of the three orthographic views, and the correspondence between the edges in the* $3D$ *edge loop and the* $2D$ *edges in the* $2D$ *loop is one-to-one.*

Proof. Edge loops are boundaries of 3D faces. They can be divided into two classes according to the types of the faces: the edge loop in the planar face and the edge loop in the quadric or toroidal surface.

A plane cannot be perpendicular to three coordinate planes which are perpendicular to each other simultaneously, therefore each line edge in an edge loop which belongs to a planar face is not perpendicular to three coordinate planes simultaneously. By Property 1, its projection is a 2D loop in at least one view and the correspondence between the edges in the edge loop and the 2D edges in the 2D loop is one-to-one.

The projection of the conic edge onto each view is a 2D edge, so we only consider the line edge in the quadric surface (the boundaries of a toroidal face are all conic edges). The line edges in the quadric surface are either parallel to each other (cylindrical surface) or intersectional at one common 3D point and the angle between two of them is less than 90° (conical surface). Thus, at most in one projection plane, there will exist some edges which are perpendicular to this plane. That is to say, their projections are 2D edges in at least two of the three views. Therefore, for an edge loop in a quadric or toroidal face, at least two of its projections onto the orthographic views correspond to 2D loops. From Property 1, it follows that the correspondence between the 3D edges in the edge loop and the 2D edges in the 2D loop is one-to-one. □

The method we use to automatically trace edge loops in the wire-frame is called the *projection-oriented* method, which is based on Theorem 1. In this method, we first find out all 2D loops in each view using the *maximum turning angle* (MTA) method[14]. The basic idea of the MTA method is to choose a convex vertex as the starting vertex, and to select an edge that is not referred to or referred to only once as the starting edge, then search the next edge by adjacency relationships. When there is more than one edge that satisfies the surface equation, we choose the edge with the maximum turning angle. The turning angle is the angle by which the tangent vector of a curved edge is to be rotated at the common vertex to go from the tangent vector of the current curved edge to the tangent vector of its adjacent curved edge. Then, for each 2D loop identified, two 2D edges which share a common 2D vertex are chosen. Next, we select two c-edges that are generated from these two 2D edges, respectively, and make sure that the selected two c-edges also share a common c-vertex. The types of

these two edges and their relationship will determine the type of the surface to be generated (If the surface has been generated, the new edge loop will be added to this surface). In our reconstruction algorithm, for each 2D edge, the indices of the c-edges that are generated from it are stored as an attribute of that 2D edge. This greatly reduces the searching space. Finally, the algorithm searches for the other edges of the edge loop from the 3D edges generated from the 2D loop. This exploration is a depth first search, which begins with the vertices on the surface. For each vertex on the surface, only the edges connected to this vertex and generated from the 2D loop are checked. All the edge loops that lie in the same surface are linked together to make the final faces. Before constructing the 3D c-faces, the edge loops with inner edges subdividing them are removed from the surfaces.

Fig.2 illustrates the generation of edge loops from the 2D loop. Fig.2(a) shows a 2D loop $C(f_A f_B) \to C(f_B f_C) \to C(f_C f_D) \to C(f_D f_A)$ in the front view. Vertices $V_A(V'_A)$, $V_B(V'_B)$, $V_C(V'_C)$, and $V_D(V'_D)$ are generated from f_A, f_B, f_C, and f_D, respectively (see Fig.2(b)). For 2D edges $C(f_A f_B)$ and $C(f_B f_C)$ which share a common 2D vertex f_B, we select two c-edges which are generated from those two 2D edges, respectively; moreover, the two 3D edges share a common 3D vertex. For example, c-edges $E(V_A V_B)$ and $E(V_B V_C)$ are chosen, the direction of the two edges are the same as their projections, i.e., the current travel direction is $V_A \to V_B \to V_C$, where $V_A \to V_B$ means from V_A to V_B. Then, we search for the next edge which is adjacent to the vertex V_C from the 3D edges generated from the 2D loop. As shown in Fig.2(b), 3D edge $E(V_C V_D)$ satisfies this condition. Next, 3D edge $E(V_D V_A)$ is selected and an edge loop is traced. Similarly, we can find 3D edge loop $E(V'_A V'_B) \to E(V'_B V'_C) \to E(V'_C V'_D) \to E(V'_D V'_A)$ (see Fig.2(c)).

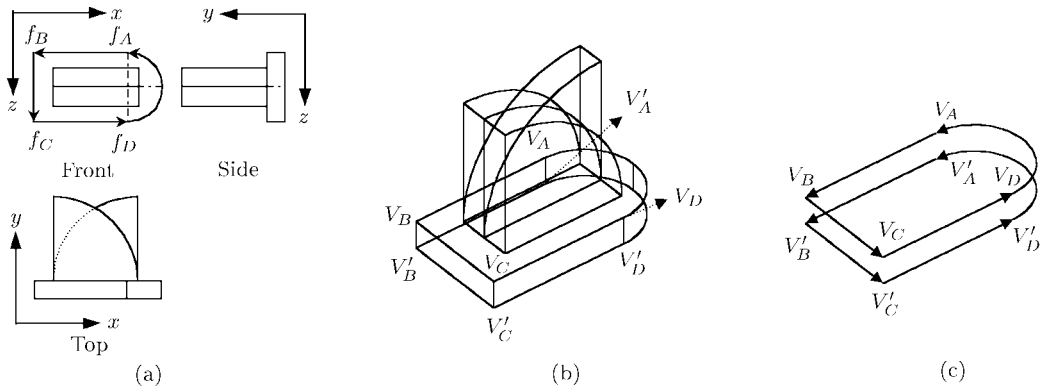

Fig.2. Trace edge loops in the wire-frame.

In the previous method, the combination of c-edges is performed $n(n-1)/2$ times in generating all the surfaces, where n is the number of c-edges in the wire-frame. The number of times for tracing c-edges which belong to a surface is n. While in our method, the combination of c-edges is performed $\tilde{c}^2 m$ times, where $\tilde{c}$ is the average number of c-edges generated from a 2D edge, m is the number of 2D edge loops in the views. Since $m \ll n_v$, n_v is the number of 2D edges in the views, and $\tilde{c}m \ll \tilde{c}n_v - 1 = n - 1$, $\tilde{c} < n/9 < n/2$ (each view consists of at least three 2D edges), therefore $\tilde{c}^2 m \ll n(n-1)/2$. The number of times for tracing c-edges in a surface is $\tilde{c}m_l$, m_l is the number of 2D edges in the 2D loop. Since $m_l \ll n_v$, it follows that $\tilde{c}m_l \ll n$. Therefore, the processing time required for generating c-faces is reduced noticeably.

4 Implementation

After developing the method, it is necessary to check it by some real mechanical parts and consider their reconstruction from orthographic views.

Fig.3 shows a mechanical part with planar, conical, toroidal, and cylindrical faces. Fig.4 shows a unique solution constructed from a three-view engineering drawing with straight lines, circular arcs,

and elliptical arcs. The axes of the cylindrical surfaces that intersect with the top planar face (see from the positive direction of z-axis) are not parallel to any of the three coordinate axes.

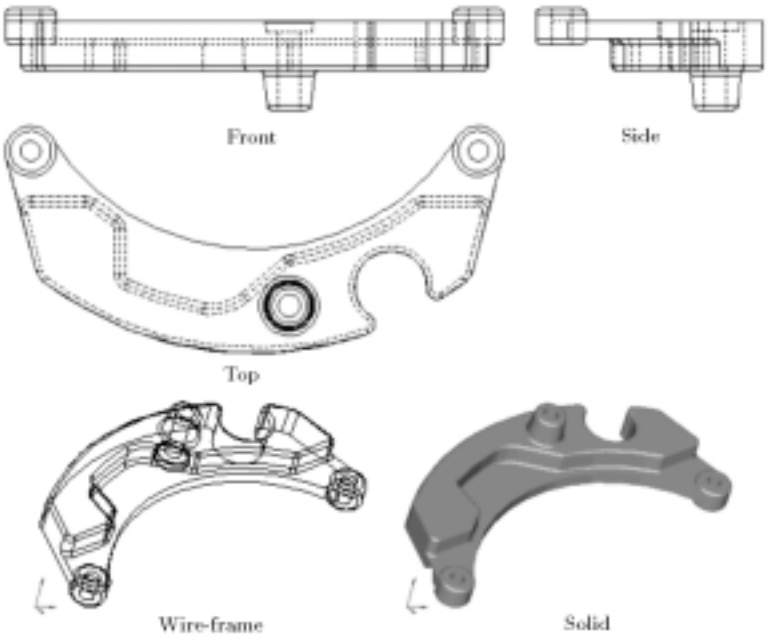

Fig.3. Engineering drawings and reconstructed model for object 1.

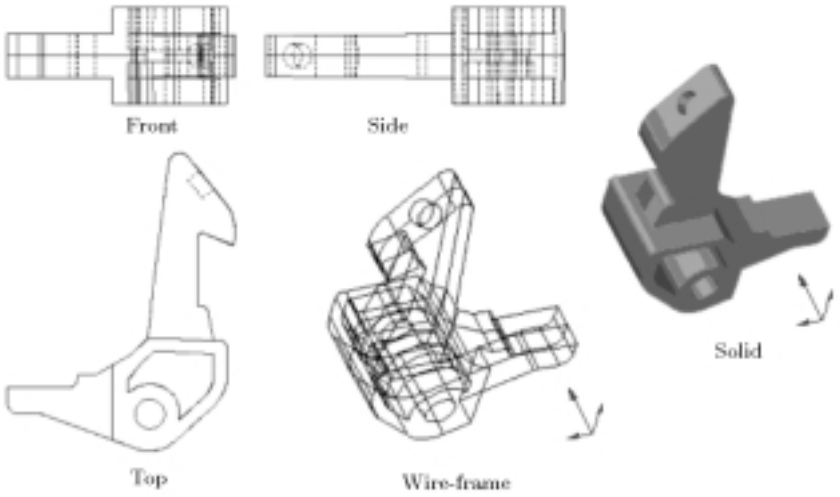

Fig.4. Engineering drawings and reconstructed model for object 2.

5 Conclusion

Reconstructing 3D objects from orthographic views requires a considerably amount of processing time. In this paper, we describe two accelerating techniques to improve the efficiency of reconstruction. One reduces the searching space by removing some pseudo elements in the wire-frame directly; while the other accelerates the generation of 3D faces by using the correspondence between the 3D face and its projections.

Our future work includes making full use of the annotation in engineering drawings to extend the

domain of objects to be reconstructed.

Acknowledgements We would like to thank Professor Chen Yujian for her constant encouragement and advice.

References

[1] Nagendra I V, Gujar U G. 3-D objects from 2-D orthographic views—A survey. *Computer and Graphics,* 1988, 12(1): 111–114.
[2] Wang W, Grinstein G G. A survey of 3D solid reconstruction from 2D projection line drawings. *Computer Graphics Forum,* 1993, 12(2): 137–158.
[3] Chen Z, Perng D B. Automatic reconstruction of 3D solid objects from 2D orthographic views. *Pattern Recognition,* 1988, 21(5): 439–449.
[4] Meeran S, Pratt M J. Automated feature recognition from 2D drawings. *Computer-Aided Design,* 1993, 25(1): 7–17.
[5] Masuda Hiroshi, Numao Masayuki. A cell-based approach for generating solid objects from orthographic projections, *Computer-Aided Design,* 1997, 29(3): 177–187.
[6] Shum S P, Lau W S, Yuen M F, Yu K M. Solid reconstruction from orthographic opaque views using incremental extrusion. *Computer and Graphics,* 1997, 29(6): 787–800.
[7] Shum S P, Lau W S, Yuen M F, Yu K M. Solid reconstruction from orthographic opaque views using 2-stage extrusion. *Computer-Aided Design,* 2001, 33(1): 91–102.
[8] Idesawa M. A system to generate a solid figure from three views. *Bulletin of the JSME,* 1973, 16(92): 216–225.
[9] Markowsky G, Wesley M A. Fleshing out wire frames. *IBM J. RES. DEVELOP,* 1980, 24(5): 582–597.
[10] Wesley M A, Markowsky G. Fleshing out projection. *IBM J. RES. DEVELOP,* 1981, 25(6): 934–954.
[11] Sakurai H, Gossard D C. Solid model input through orthographic views. In *SIGGRAPH'83 Proceedings,* 1983, pp.243–252.
[12] Kuo M H. Reconstruction of quadric surface solids from three-view engineering drawings. *Computer Aided Design,* 1998, 30(7): 517–527.
[13] Oh B S, Kim C H. Systematic reconstruction of 3D curvilinear objects from two-view drawings. *Computers & Graphics,* 1999, 23(3): 343–352.
[14] Liu S X, Hu S M, Chen Y J, Sun J G. Reconstruction of curved solids from engineering drawings. *Computer-Aided Design,* 2001, 33(14): 1059–1072.
[15] Liu S X, Hu S M, Tai C L, Sun J G. A matrix-based approach to reconstruction of 3D objects from three orthographic views. In *Proceedings of the Eighth Pacific Conference on Computer Graphics and Applications,* Barsky B A, Shinagawa Y, Wang W (eds.), Hong Kong: IEEE Computer Society, 2000, pp.254–261.