Reconstruction of 3D Interacting Solids of Revolution from 2D Orthographic Views

Hanmin Lee, Soonhung Han

Department of Mechanical Engeneering

Korea Advanced Institute of Science & Technology

373-1, Guseong-Dong, Yuseong-Gu,

Daejeon 305-701, Korea

Tel : +82-42-869-3080

Fax : +82-42-869-3210

komaa@icad.kaist.ac.kr, shhan@kaist.ac.kr

Abstract

3D CAD is replacing 2D CAD to improve efficiency of product design and manufacturing. Therefore, converting legacy 2D drawings into 3D solid models is required. CSG based approaches reconstruct solid models from orthographic views more efficiently than traditional B-rep based approaches. A major limitation of CSG based approaches has been the limited domain of objects that can be handled. This paper aims at extending the capabilities of CSG based approaches by proposing a hint-based recognition of interacting solids of revolution. This approach can handle interacting solids of revolution as well as isolated solids of revolution.

Keywords: Solid Reconstruction, Interacting Solids of Revolution, Hint-based Recognition, Orthographic Projections

1. Introduction

Two-dimensional engineering drawings have been a staple of the design process since the late eighteenth century, when the principles of orthographic projection and descriptive geometry were first developed and applied to engineering problems. 2D drawings still play an important role in engineering practice, and in many cases serve as the definitive design documentation that guides manufacture, fabrication, and assembly of products. However, 2D drawings have critical limitations that unnecessarily extend design cycle time, compromise product quality, and increase engineering and manufacturing costs. These weaknesses primarily arise from the aspects that it is difficult to inspect 2D design data intuitively and to verify it without physical prototyping and to use it directly in downstream processes.

It is necessary to convert existing 2D drawings into 3D solid models, because 3D solid models can overcome the limitations of 2D drawings. There have been two general approaches for solid reconstruction according to the solid representation schemes used: B-rep based approach and CSG based approach [6, 7]. The B-rep based approach has a relatively wider domain of objects that can be reconstructed than the CSG based approach. However, it cannot reconstruct complex objects efficiently due to exhaustive search and computation, and ambiguity may occur. The CSG based approach is relatively more efficient and often provides a unique solution, because it generally employs Boolean operations that ensure the validity of geometric models, thus avoiding the creation of nonsense objects. However, it has a limited domain of objects because it typically uses either pre-defined 2.5D primitives or identifies only entities that can be extruded.

The purpose of this paper is to extend the capabilities of the CSG based approach by proposing a procedure to reconstruct solids of revolution from orthographic views. A hint-based method is proposed to recognize interacting solids of revolution as well as isolated ones using minimal conditions. This approach is restricted to recognizing axis-aligned solids of revolution (axis of revolution is normal to one of the planes of projection).

2. Related Work

2.1. B-rep based approach

Most of the work done on constructing solid models from orthographic views has been based on the B-rep based approach. Fig. 1 shows the typical procedure of the B-rep based approach. The B-rep based approach assembles candidate vertices, edges, and faces. The B-rep based approach was first proposed by Idesawa[1]. The most comprehensive work based on this approach has been done by Markowsky[2] and Wesley[3], who introduced the concept of constructing an intermediate wireframe. Sakurai[4], Yan[8], Shin[9], Kuo[10], and Liu[14] have extended the work of Markowsy and Wesley to develop more efficient, precise, and robust algorithms with a wider input geometric domain.

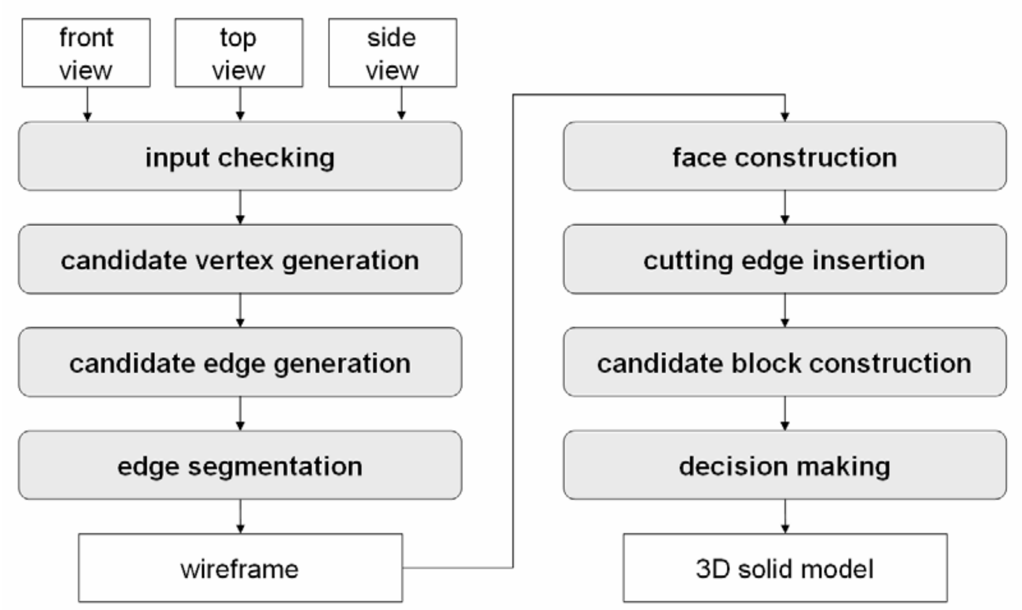

Fig. 1. A typical pipeline of B-rep based approach

2.2. CSG based approach

The CSG based approach creates elementary solids by recognizing patterns in 2D projections and assembles them to obtain the solution. Aldefeld [5] interpreted the orthographic views based on pattern recognition where the patterns being searched for are projections of pre-defined solid primitives. This approach is restricted to objects with uniform thickness, and requires user intervention in the pattern recognition process. Shum [15] proposed a 2-stage extrusion method. Fig. 2 shows the procedure of 2-stage extrusion method. In the first stage an exterior contour region in each view is swept along its normal direction, and a bounding solid (called a *basic-solid*) is generated using the intersection of the three swept solid (called an *extrusion-solid*). Next, interior entities in each view are filtered and processed to generate an internal solid (called an *excess-solid*). Subtracting the excess-solid from the basic-solid generates the final 3D solid.

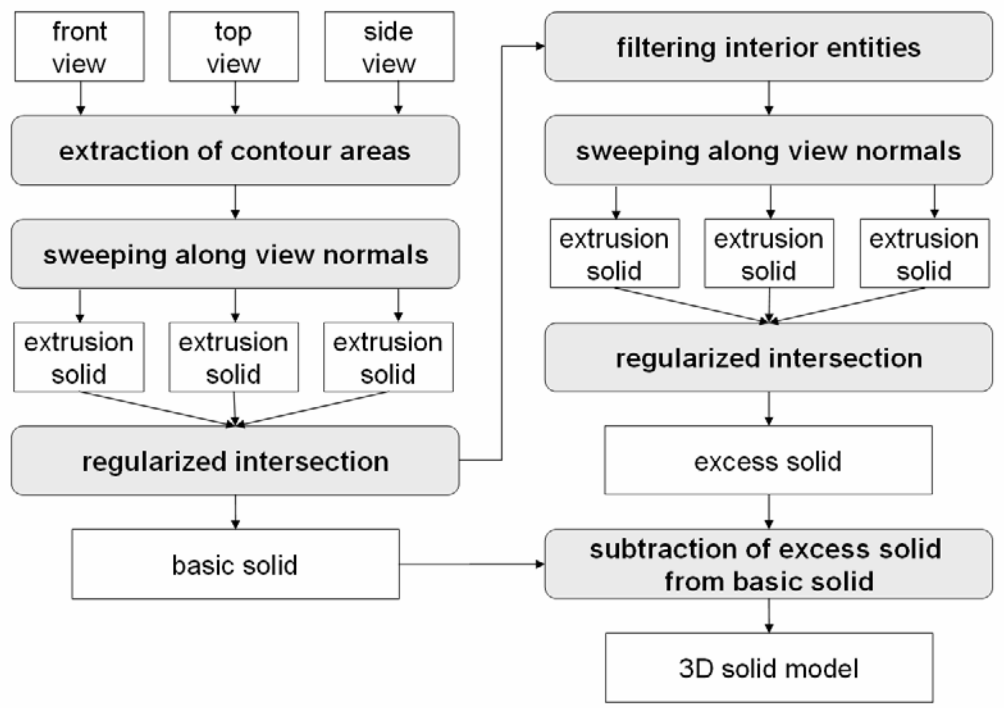

Fig. 2. Two-stage extrusion algorithm[15]

Soni [17] proposed a procedure to identify entities in the views that correspond to axis-aligned volumes of revolution and then use these to construct the corresponding primitive volumes. Soni's approach extended the domain of objects by reconstructing solids of revolution that cannot be handled by previous CSG based approaches. However, criteria for identifying entities corresponding to solids of revolution are limited to simple cases where there is no interaction between the volumes. For example, Fig. 3 depicts views of a spherical part that has some region of the sphere removed. In this case, the entities in the views do not satisfy the conditions for rotational sweep, and therefore Soni's approach cannot regenerate the spherical part from 2D projections.

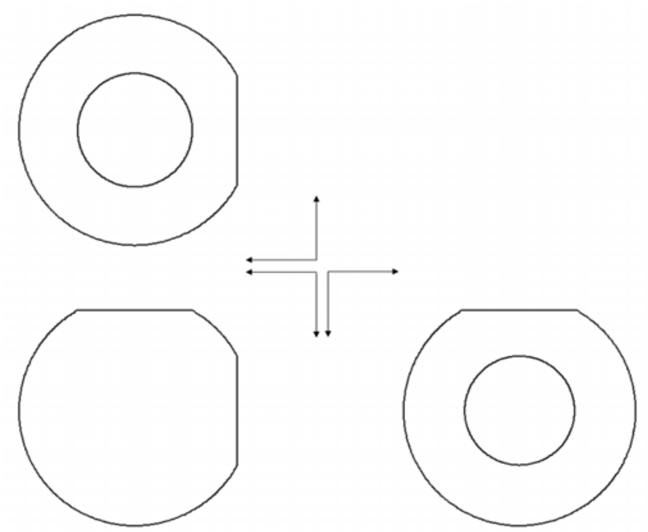

Fig. 3. Spherical part with removed portions[17]

We propose a hint-based method that recognizes interacting solids of revolution from orthographic views. Isolated solids of revolution without interaction are generated from minimal hints. Subtracting the interacting volumes from the intact solids of revolution, as shown in Fig. 3, generates interacting solids of revolution. Using this method, we have extended the domain of objects that can be handled by the CSG based approach.

3. Terminology

Common coordinate axis: The coordinate axis that is common between a pair of views. In Fig. 4, the *x-axis* is the common coordinate axis between *view 1* and *view 2*.

Loop: A simple closed cycle of edges in a view is defined as a loop. An endpoint of each edge can be connected by two and only two edges.

Extreme points: Points on a circle with the maximum or minimum coordinate values along the common coordinate axis are called as the extreme points of the circle. If the point p on the circle C has the maximum coordinate value along the common coordinate axis, p is called as the maximum point of C. If the point p on the circle C has the minimum coordinate value along the common coordinate axis, p is called as the minimum point of C. In Fig.4, p_1 is the maximum point of C_1, and p_4 is the minimum point of C_1 along the common coordinate *x-axis*.

Matching vertices: The vertex V in the view G_1 is said to be matching with the circle C in the view G_2, if V has the same coordinate value as one of the extreme points of C along the common coordinate axis. If V has the same coordinate value as the maximum point of C, V is called maximum matching vertex. If V has the same coordinate value as the minimum point of C, V is called minimum matching vertex. For example, in Fig. 4, the maximum matching vertices of C_1 are v_1 and v_2, and the minimum matching vertices of C_1 are v_6 and v_7.

Matching edges: The edge E in the view G_1 is said to be matching with the circle C_1 in the view G_2 if E is one of the three types whose definitions are given below:

Type I: both of the end vertices of E are the maximum matching vertices (or minimum matching vertices) of C_1, as shown in Fig. 8a

Type II: one of the end vertices of E is the matching vertex of C_1, and the other is on the axis of rotation, as shown in Fig. 8b

Type III: one of the end vertices of E is the maximum matching vertex (or minimum matching vertex) of C_1, and the other is the maximum matching vertex (or minimum matching vertex) of C_2 that is smaller than C_1 and concentric with C_1, as shown in Fig. 8c.

In Fig. 4, for example, e_1 and e_6 are *Type I* matching edges of C_1, and e_2 and e_5 are *Type III* matching edges of C_1 , and e_3 and e_4 are *Type II* matching edges of C_2.

Virtual region: A region that is formed by subtracting original loops from new loops that are generated to make the complete patterns for isolated solids of revolution before interaction on the assumption that some parts of the solids of revolution are cut off. The virtual region in the view that is perpendicular to the axis of rotation, as shown in Fig. 7b, is expressed as R_t, and in the view that is parallel to the axis of rotation, as shown in Fig.11, is expressed as R_s.

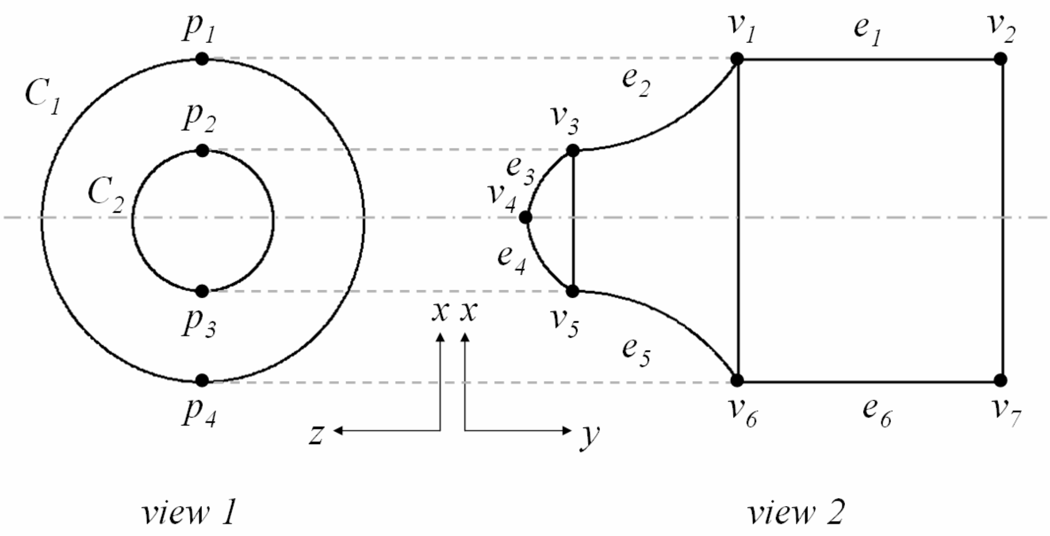

Fig. 4. Matching vertices and matching edges

4. The patterns for solids of revolution in orthographic views

If a solid that is generated by rotating a profile 360° along an axis is projected to 2D planes, as Fig. 5a shows, there should be circles on the view that is perpendicular to the axis of rotation, and entities that are symmetric with each other along the axis of rotation on the view that is parallel to the axis of rotation. Using these patterns for solids of revolution, the existing approach can handle isolated solids of revolution that have no interaction with other volumes.

However, if some parts of solids of revolution are cut off, or the solid is generated by rotating a profile less than 360°, it is not possible to identify entities using the above basic patterns. Fig. 5b indicates that in the case of interacting solids of revolution, there are circular edges that are not complete circles in the top view, and entities that are not fully symmetric with each other along the axis of rotation in the side view.

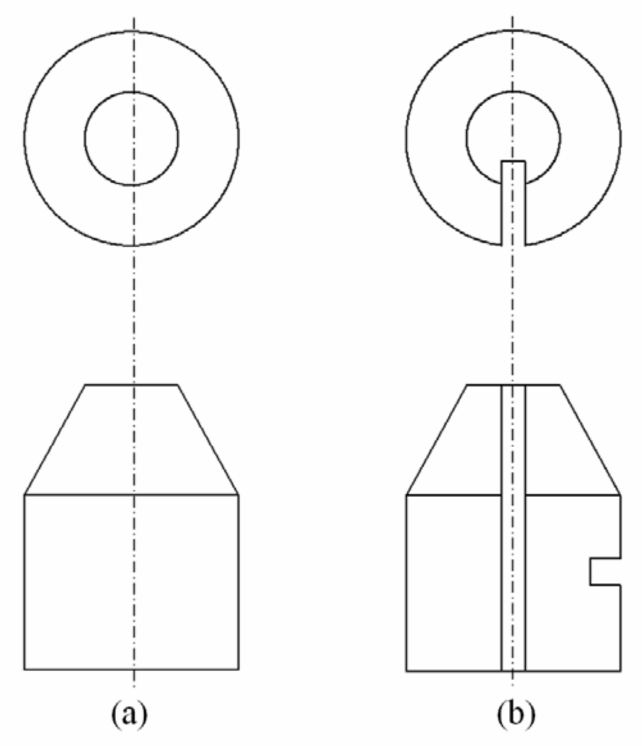

Fig. 5. 2D pattern for (a) isolated and (b) interacting volume of revolution

To handle such interacting solids of revolution that cannot be reconstructed by using the basic conditions, we propose a hint-based method, which can construct solids of revolution from the traces of pattern for solids of revolution. Fig. 6 shows the procedure of the hint-based reconstruction method for solids of revolution. Hints such as a circle or arc are searched in a view, and entities matching with the hints are searched in other views. Through tests for rotational symmetry between the matching entities, rotational profiles are generated. Revolving the profiles around the axis of rotation makes complete volumes of revolution without the interaction. Subsequently, virtual regions indicating the volumes that should be cut off are extruded, and the volumes are subtracted from the complete solids to make the final interacting solid of revolution.

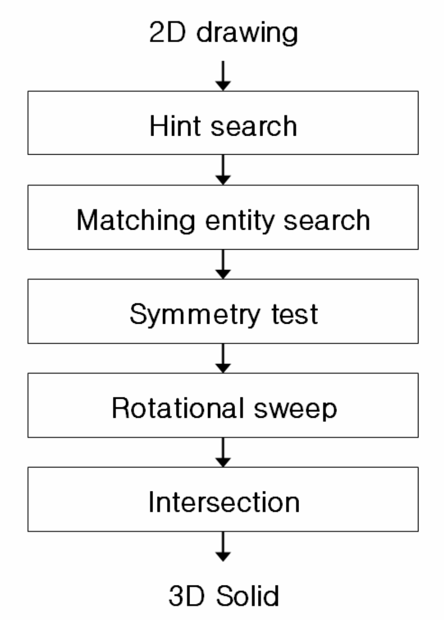

Fig. 6. Hint-based solid reconstruction process

5. Hint-based method for recognizing solids of revolution

Input drawings are in DXF(Drawing Interchange Format) file format. By parsing the input drawings according to the group codes and group values of DXF, 2D geometric entities such as *LINE*, *ARC*, and *CIRCLE* are identified. These entities are converted as a graph which consists of a set of vertices, edges, and incidence relations[13].

5.1. Hint search

Circular edges are the most important clues for recognizing solids of revolution in 2D views. In the case of projection of interacting solids of revolution, there will not always be complete circles in the top view. Interaction with other volumes causes the circles to be broken in the projections. On the assumption that an arc implies there has been an interaction between the solid of revolution and other volumes, the arc is extended to a circle, which is the basic pattern of isolated solids of revolution. To search the profiles along the same axis of rotation, the circles that have same center are inserted into the concentric circle list. According to the above procedure, circular edges are searched as follows:

(1) Find all the circles in the views, and insert them into the circle lists.

(2) Construct circles from arcs in the views, and insert them into the circle lists.

(2.1) For an arc *A*, construct circle *C* using the center point and radius of *A*.

(2.2) For the entities inside *C*, make the outer loop *L* by connecting the adjacent edges starting from *A* using FLG (face-loop generation) algorithm, as described in [8].

(2.3) If a closed loop *L* is generated, subtract loop *L* from circle *C* to form the virtual region in the top view, R_t, and insert circle *C* into the circle lists. Otherwise, stop testing for arc *A*.

(3) Select the circles that have the same center from the circle lists, and insert them into the concentric circle lists.

For example, the hints for solids of revolution are arc A_1 and circle C_1 in Fig. 7a. As Fig. 7b shows, arc A_1 is extended to make circle C_2, and the virtual region R_t is formed during this process.

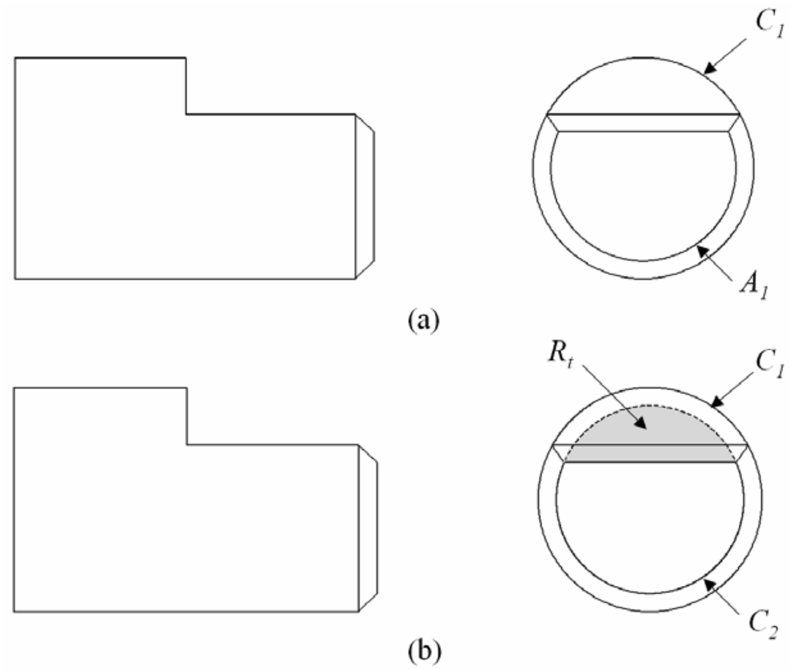

Fig. 7. Hint search

5.2. Matching entity search

Vertices and edges in other views matching the circle in the previous view are found by checking the coordinate along the common coordinate axis. Three basic types of matching edges of the circle are defined in this paper:

Type I: the edge that connects two matching vertices of the circle, Fig. 8a.

Type II: the edge that connects one matching vertex of the circle and one vertex on the rotational axis, Fig. 8b.

Type III: the edge that connects two matching vertices of two concentric circles, Fig. 8c.

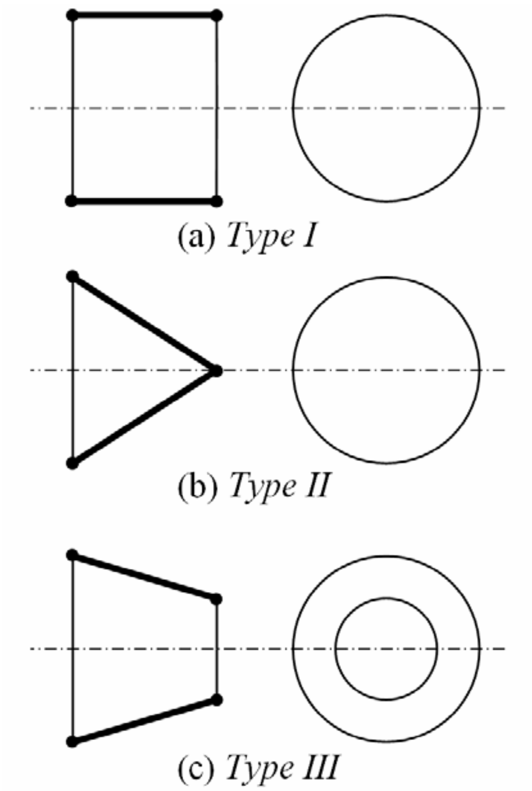

Fig. 8. Types of matching edge

Matching vertices and matching edges are searched as follows:

(1) Matching vertex search

For every pair of views, G_i and G_j (i≠j),

(1.1) Find common coordinate axis α between G_i and G_j.

(1.2) Find vertex V in G_j whose α coordinate value is the same as that of the maximum point of circle C in G_i, and insert V into the maximum matching vertex list $VList_{max}$ of C. In the same manner, find the minimum matching vertex of C and insert it into the minimum matching vertex list $VList_{min}$ of C.

(2) Matching edge search

For each circle C_i in concentric circle list, (2.1) For every pair of maximum matching vertices, V_k and V_{k+1} in $VList_{max}$ of C_i, find edge E that connects V_k and V_{k+1} and insert E to the maximum matching edge list $EList_{max}$ of C_i. In the same manner, find the edge that connects two minimum matching vertices of C_i and insert it into the minimum matching edge list $EList_{min}$ of C_i. (*Type I*)

(2.2) Find edge E that connects V_k in $VList_{max}$ and vertex V that is located on the axis of rotation, and insert E into $EList_{max}$ of C_i. In the same manner, find the edge that connects a minimum matching vertex and a vertex located on the axis of rotation, and insert it into $EList_{min}$ of C_i. (*Type II*)

(2.3) For circle C_j (j=0,1,…,i-1) which is smaller than C_i in the concentric circle list, find edge E that connects a vertex in $VList_{max}$ of C_i and a vertex in $VList_{max}$ of C_j, and insert E into $EList_{max}$ of C_i. In the same manner, find the edge that connects a minimum matching vertex C_i and a minimum matching vertex of C_j and insert it into $EList_{min}$ of C_i. (*Type III*)

Fig. 9 shows the matching vertices and edges of the circles. The common coordinate axis between *view 1* and *view 2* is the *x-axis*. Searching vertices in *view 1* matching with circle C_1 in *view 2* along the common coordinate axis, the maximum matching vertex list $VList_{max}$ = {*8, 7*} and the minimum matching vertex list $VList_{min}$ = {*1, 2*}. For circle C_2, the minimum matching vertex is vertex *3*, and there is no maximum matching vertex of C_2 in the views. Searching edges in *view 1* matching with circle C_1, edges *1-2* and *8-7* are found by the condition (2.1) given above, and edge *2-3* is found by the condition (2.3) given above. For circle C_2, there is no edge that satisfies the conditions for a matching edge.

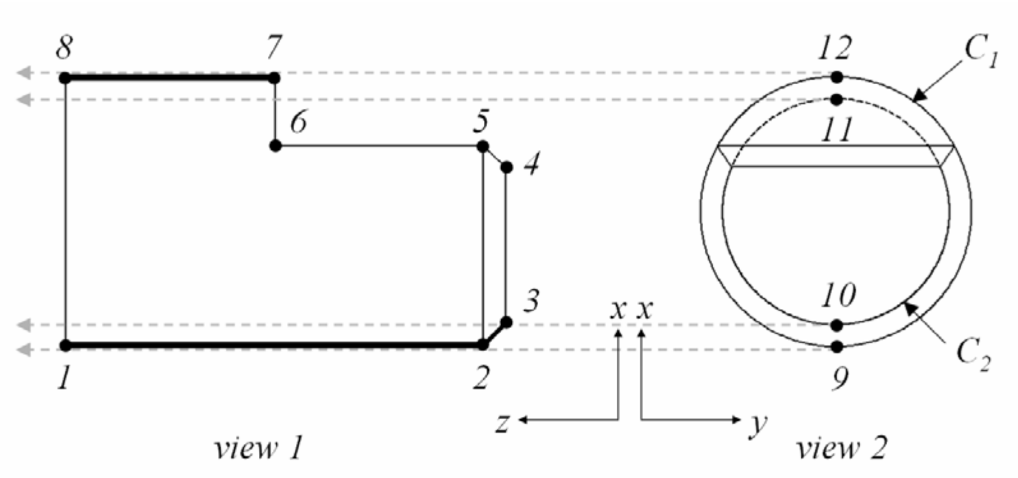

Fig. 9. Matching entity search

5.3. Symmetry test

To generate the rotational profiles the matching edges of each circle should be tested against rotational symmetry. If two matching edges of a circle are symmetric with each other along the axis of rotation, they can be a part of the profiles. However, in the case of interacting solids of revolution, the matching edges do not always satisfy the rotational symmetry. Interaction with other volumes breaks the rotational symmetry in the projections. Only a portion of the edges can satisfy the rotational symmetry, and the edges symmetric with other edges can even disappear in the projections. The following algorithm considers such exceptional cases.

For $EList_{max}$ and $EList_{min}$ of each circle C,

(1) Test for full symmetry

(1.1) If a maximum matching edge E_{max} in $EList_{max}$ and a minimum matching edge E_{min} in $EList_{min}$ are symmetric with respect to the axis of rotation, insert the pair of E_{max} and E_{min} into the profile list $PList$ of C.

(2) Test for partial symmetry

(2.1) If a part of E_{max} and a part of E_{min} are symmetric with respect to the axis of rotation, extend them to make E_{max}' and E_{min}' that are fully symmetric with respect to the axis of rotation.

(2.2) Insert E_{max}' and E_{min}' into *PList*, and obtain the virtual region, R_s, formed by the extended part of E_{max}' and E_{min}'.

(3) Test for removed edge

(3.1) If, for each maximum matching edge E_{max}, there is no edge that satisfies conditions (2) and (3) given above, check if circle C is generated from an arc in the views.

(3.2) If C is generated from an arc, which does not have the minimum point of C, generate E_{min}* that is the symmetric edge of E_{max} with respect to the axis of rotation.

(3.3) Insert E_{max} and E_{min}* into *PList*, and obtain the virtual region, R_s, formed by E_{min}*.

(3.4) In the same manner, execute (3.1), (3.2), and (3.3) given above for E_{min} in $EList_{min}$.

Fig. 10 shows the process that generates the rotational profiles through the symmetry test for matching edges. In Fig. 10a, two matching edges are symmetric with respect to the axis of rotation. Therefore, they can be a profile without any modification. In Fig. 10b, a part of two matching edges is symmetric with respect to the axis of rotation. In this case, we can expect that a part of the rotational profiles has been removed due to the interaction with other volume. Fig. 10c shows the rotational profile generated by extending the partially symmetric edges. Fig. 10d shows a case where an edge of the matching edge pair is entirely missing. If we can determine that the interaction with other volume has removed the symmetric edge through the condition (3) given above, the symmetric edge must be generated to recover the correct shape, as shown in Fig. 10e. The virtual region that is generated by the virtual edge in Fig. 10c and Fig. 10e must be swept and subtracted from the complete volume of rotation.

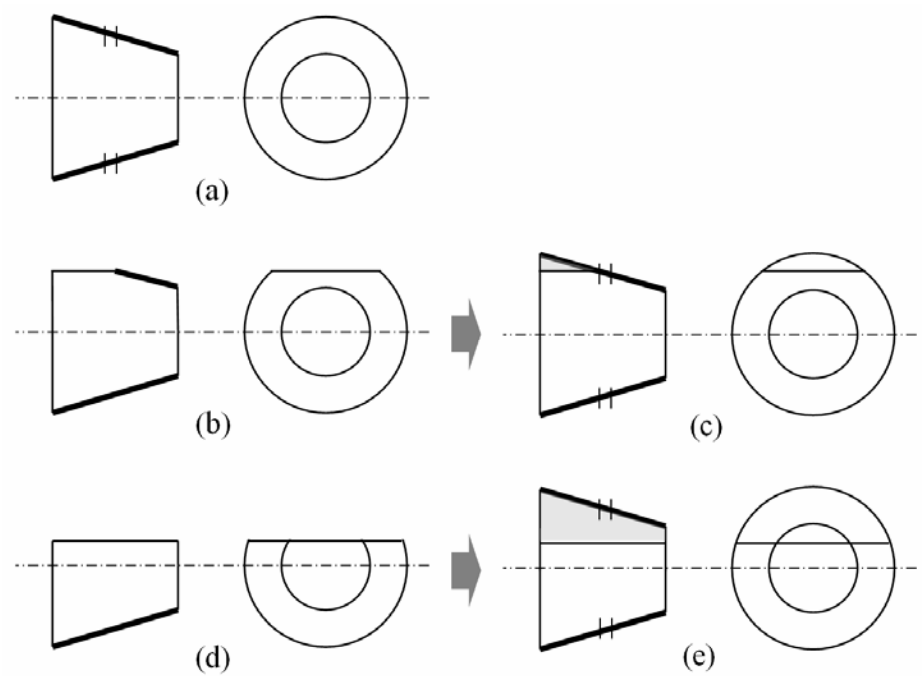

Fig. 10. Symmetry test

In Fig. 11, edges *1-2* and *8-7* are partially symmetric with respect to the axis of rotation, and edge *8-7* is extended to make edge *8-13* such that edges *1-2* and *8-13* are symmetric. Edge *2-3* does not have a symmetric edge. However, the condition (3) given above is satisfied in this case, and hence edge *13-14* is newly generated to be the symmetric edge of edge *2-3*. The virtual region, R_s, is generated during these processes.

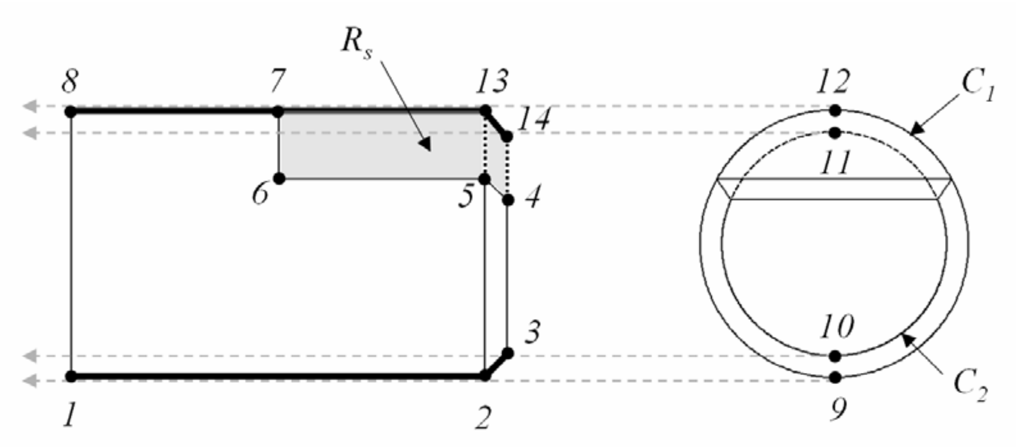

Fig. 11. Rotational profile generation through symmetry test

5.4. Rotational sweep and intersection

In this process, revolution of rotational profiles which are found in the previous process generates complete solids of revolution that have no interaction. Then the virtual regions are generated and subtracted from the complete volumes of revolution.

(1) Revolve the profiles in *PList* along the axis of rotation.

(2) If there exists a virtual region R_t on circle C, and the matching edges of C is used to make rotational profiles, sweep R_t along the normal direction of the projection plane. Subtract the swept volume from the solids of revolution generated in (1).

(3) If there exists a virtual region R_s, sweep R_s along the normal direction of the projection plane, and subtract the swept volume from the solids of revolution.

Fig. 12 shows a profile to be swept and a volume of rotation generated using the profile. In Fig. 13, the virtual region R_s is swept, and the swept volume is subtracted from the original volume to make final interacting solids of revolution. As shown in Fig. 7, circle C_2 has the virtual region R_t, but there is no matching edge of C_2 that is used to make the rotational profile. Therefore, R_t of C_2 is not swept to make the volume to be subtracted.

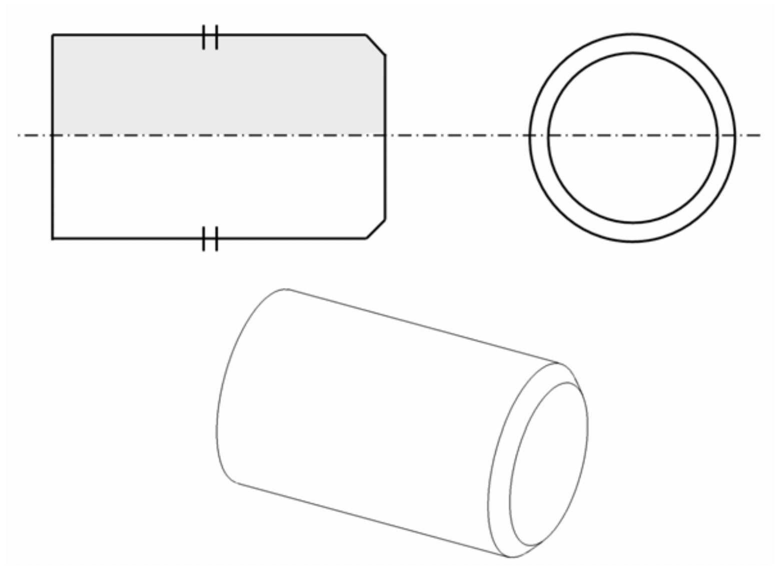

Fig. 12. Rotational profile and volume of revolution without interaction

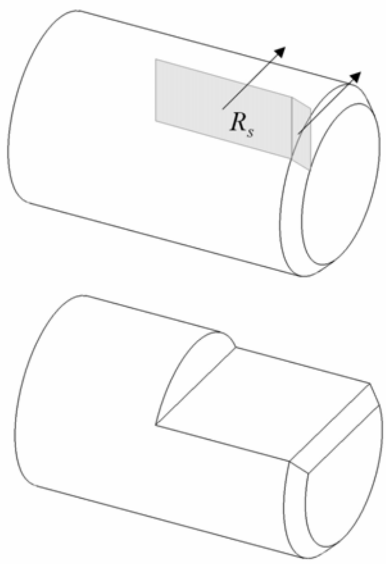

Fig. 13. Extrusion of virtual region and intersection

6. Implementation and Experiments

A hint-based reconstruction system for a rotational part has been implemented using Visual Basic 6.0 and Solidworks 2004 API. Several sample drawings are tested to verify the proposed method.

Fig. 14 shows the result of reconstructing a switch from orthographic views. Fig. 14a is the input drawing. Fig. 14b shows the solid of revolution that is generated by the method proposed in this paper, and Fig. 14c show the solids that are generated by extrusion and intersection proposed in the existing CSG based approach[15]. Fig. 14d shows the final model combining these two volumes.

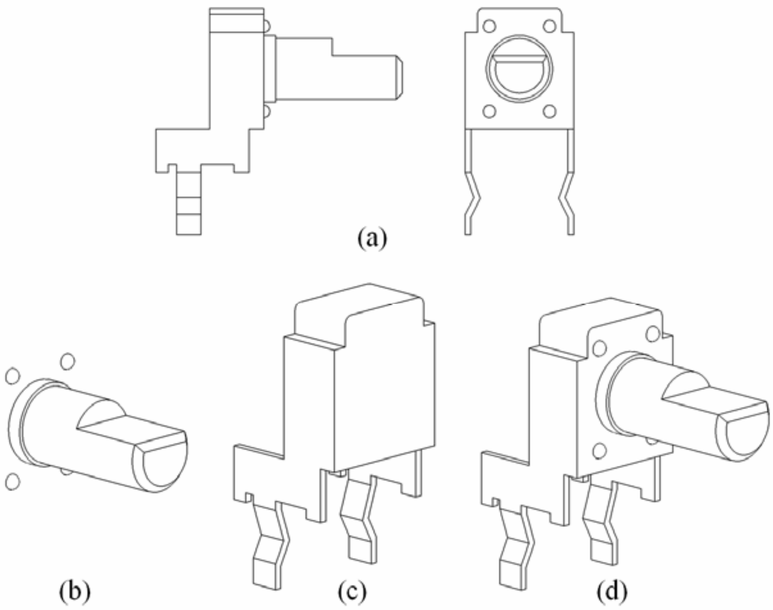

Fig. 14. Switch drawing

Fig. 15 shows the result of reconstructing a shaft. Fig. 15b shows the rotational profiles and the virtual region that are generated by *hint and matching entity search* from input drawing in Fig. 15a. Revolving the profiles generates the volumes in Fig. 15c. In Fig. 15d, the virtual region, generated in Fig. 15b, is swept and subtracted from the rotational volumes to make interacting volumes of revolution. In the same manner, a grip of screwdriver has been reconstructed in Fig. 16. Fig. 16b shows the rotational profile and the virtual region. Revolving the profile generates the volume in Fig. 16c. In Fig. 16d, the virtual region is swept and subtracted from the rotational volume to make interacting volumes of revolution.

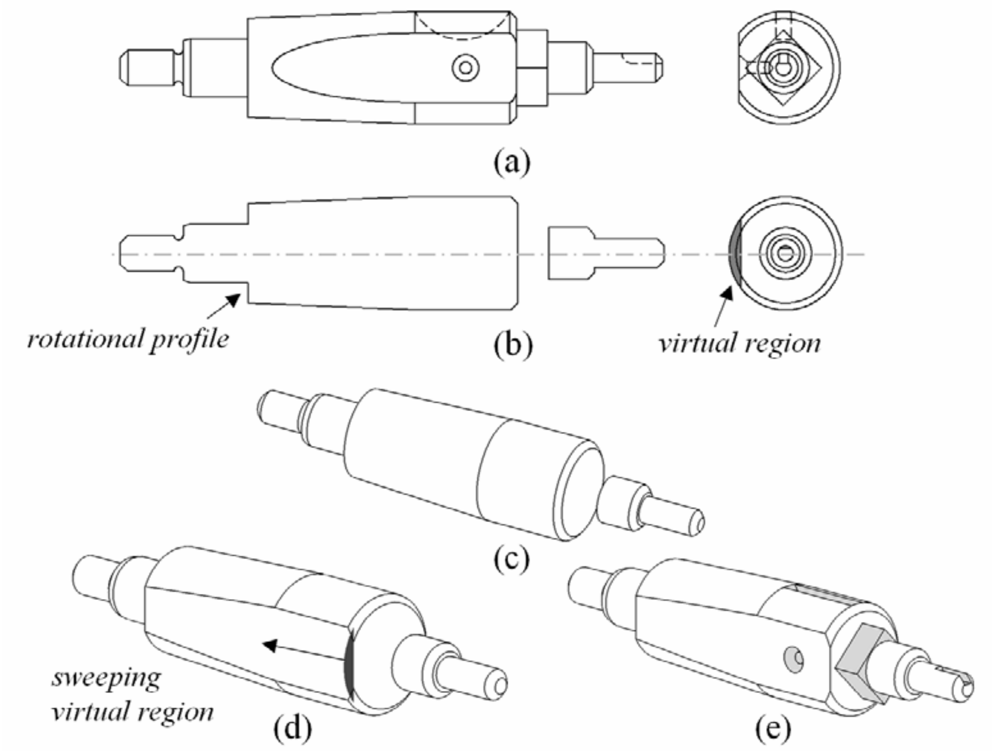

Fig. 15. Shaft drawing

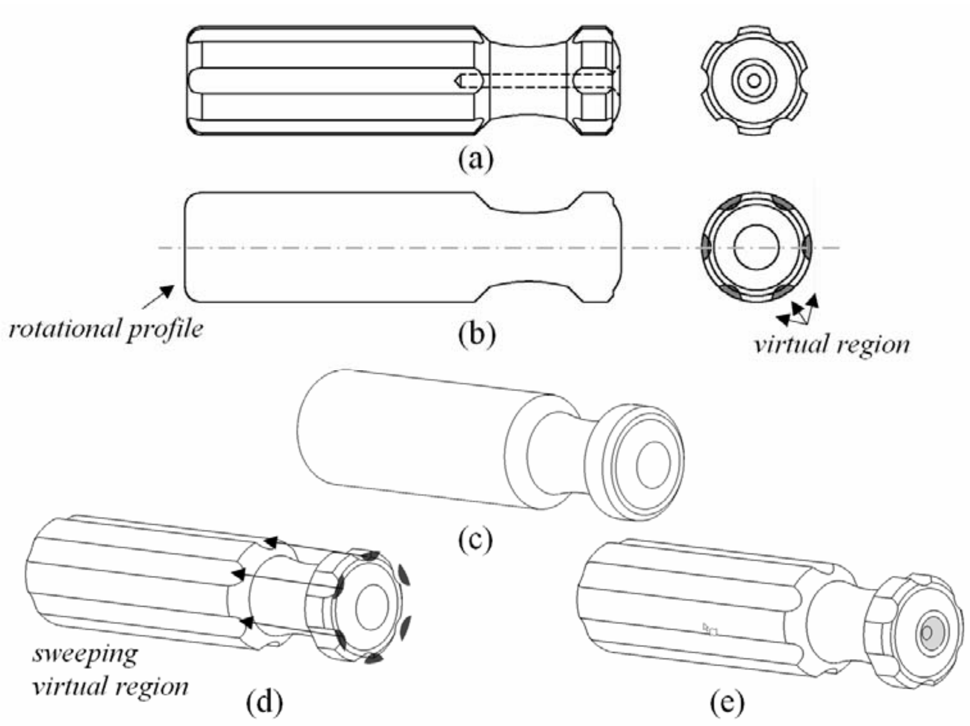

Fig. 16. Screwdriver drawing

Fig. 17 shows the reconstruction processes for a spherical part, which cannot be handled by the existing CSG based approaches. In Fig. 17a, arc *A* is taken as hint for solids of revolution in the front view. In Fig. 17b, *A* is extended to circle *C*, and virtual region R_t is formed. Now, we use the front and top view for better explanation. As shown in Fig. 17c, *C* has only one matching vertex, vertex *4* and two *Type II* matching edges, edge *3-4* and 5-*4*. Edge *3-4* and *5-4* have partially symmetric edges, edge *3-7* and *5-6*, respectively. In Fig. 17d, *symmetry test* extends edge *3-7* to *3-8* and edge *5-6* to *5-8*. During this process, the virtual region R_s is formed. Fig. 17e shows the rotational profile and the virtual region. In Fig. 17f, revolving the profile along the rotational axis makes the sphere. The virtual region is swept in Fig. 17g, and the final solid is generated by subtracting the swept volume from the sphere in Fig. 17h.

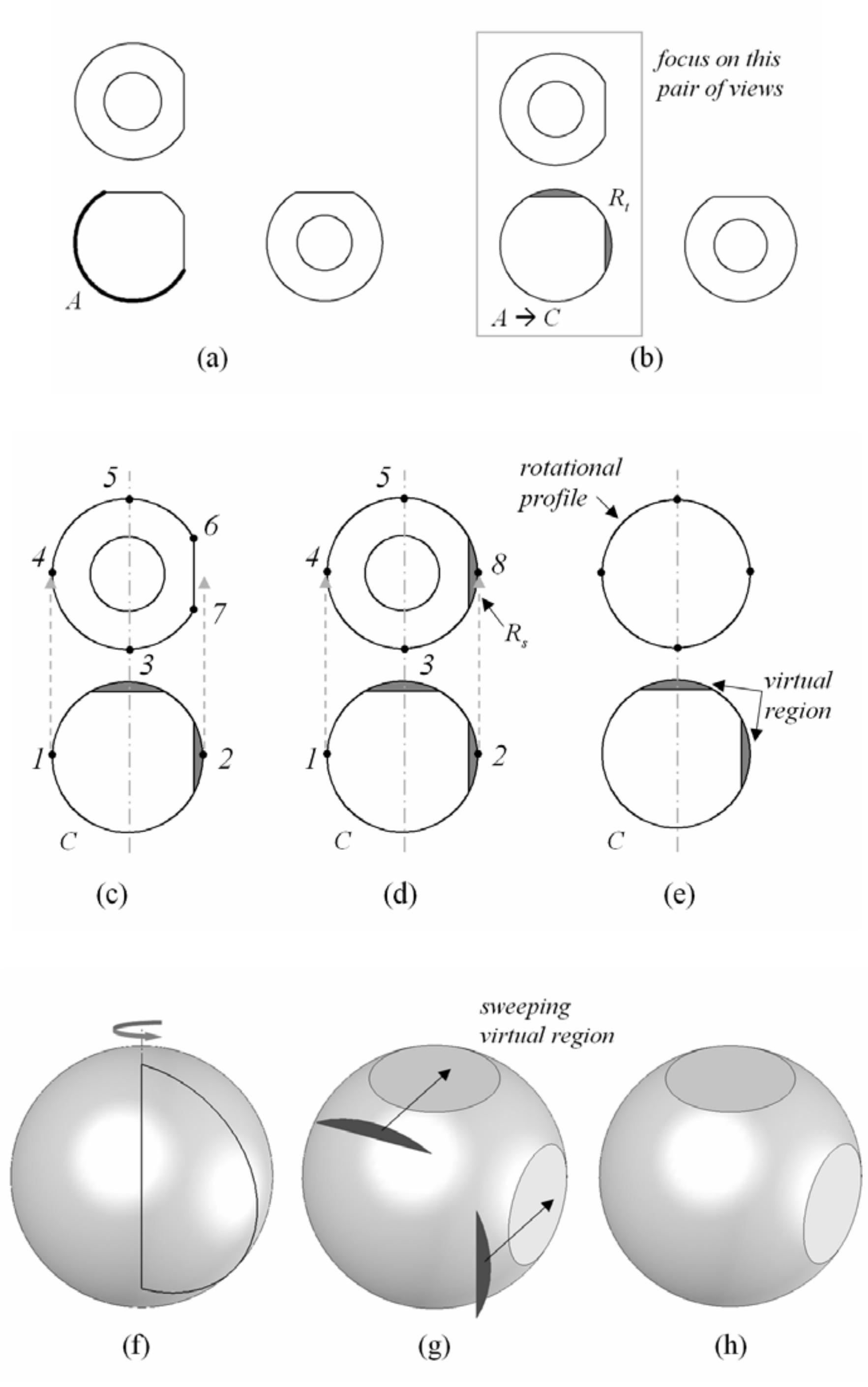

Fig. 17. Reconstruction of a spherical part

7. Conclusions

The CSG based approaches have several advantages over B-rep based approaches in that a valid solid model that is consistent with the input views is always available at any point in the solution process and there is no need for an intermediate wireframe. On the other hand, the major limitation of CSG based approaches proposed has been the limited domain of objects, because they use either pre-defined primitives or identify only entities that can be extruded. To solve this problem, an approach that can handle volumes of revolution had been proposed by Soni[17], but it can reconstruct an isolated solid that has no interaction with other volumes. In this paper, we propose an extended CSG based approach to reconstruct not only isolated but also interacting solids of revolution by using a hint-based method to identify the entities of a rotational part from the minimum traces of the pattern of solids of revolution. Several tests for sample models have been conducted to verify the effectiveness of the proposed method.

The proposed algorithm is limited to axis-aligned volumes of revolution, and the volume that intersects the complete volumes of revolution is limited to extruded solids. Reconstruction of solids of revolution which are not axis-aligned and interact with various shapes needs further work.

References

1. M. Idesawa, “A system to generate a solid figure from three views”, Bulletin of JSME, Vol.16, pp.216-225, 1973

2. G. Markowsky, M. Wesley, “Fleshing out wireframe”, IBM Journal of Research and Development, Vol.25, No.5, pp.582-597, 1980

3. M. Wesley, G. Markowsky, “Fleshing out projections”, IBM Journal of Research and Development, Vol.25, No.6, pp.934-954, 1981

4. H. Sakurai, D. Gossard, “Solid model input through orthographic views”, Computers & Graphics, Vol.17, No.3, pp.243-252, 1983

5. B. Aldefeld, “On automatic recognition of 3D structures from 2D representations”, Computer Aided Design, Vol.15, No.2, pp.59-64, 1983

6. I. V. Nagendra, V. G. Gujar, “3D objects from 2D orthographics views – a survey”, Computers & Graphics, Vol.12, No.1, pp.111-114, 1988

7. W. Wang, G. G. Grinstein, “A survey of 3D solid reconstruction from 2D projection line drawings”, Computer Graphics Forum, Vol.12, No.2, pp.137-158, 1993

8. Q. W. Yan, “Efficient algorithm for the reconstruction of 3D objects from orthographic projections”, Computer Aided Design, Vol.26, pp.699-717, 1994

9. B. S. Shin, Y. G. Shin, “Fast 3D solid model reconstruction from orthographic views”, Computer Aided Design, Vol.30, No.1, pp.63-76, 1998

10. M. H. Kuo, “Reconstruction of quadric surface solids from three-view engineering drawings”, Computer Aided Design, Vol.30, No.7, pp.517-527, 1998

11. J. Han, A. A. G. Requicha, “Feature recognition from CAD models”, IEEE Computer Graphics and Applications, Vol.18, No.2, pp.80-94, 1998

12. J. Han, W. C. Regli, S. Brooks, “Hint-based reasoning for feature recognition: status report”, Computer Aided Design, Vol.30, No.13, pp.1003-1007, 1998

13. S. Meeran, J. Taib, “A generic approach to recognizing isolated, nested and interacting features from 2D drawings”, Computer Aided Design, Vol.31, No.14, pp.891-910, 1999

14. S. Liu, “Reconstruction of curved solids from engineering drawings”, Computer Aided Design, Vol.33, pp.1059-1072, 2001

15. S. S. P. Shum, “Solid reconstruction from orthographic views using 2-stage extrusion”, Computer Aided Design, Vol.33, pp.91-102, 2001

16. W. Geng, J. Wang, Y. Zhang, ‘Embedding visual cognition in 3D reconstruction from multi-view engineering drawings’, Computer-Aided Design, Vol.34, No.4, pp.321-336, 2002

17. S. Soni, B. Gurumoorthy, “Handling solids of revolution in volume-based construction of solid models from orthographic views”, Journal of Computing and Information Science in Engineering, Vol.3, pp.250-259, 2003

18. H. Lee, S. Han, “2D design feature recognition using expert system”, Journal of Society of CAD/CAM Engineers (of Korea), Vol.5, No.2, pp.133-139, 2001

19. H. Hwang, S. Han, Y. Kim, “ Mapping 2D midship drawings into a 3D ship hull model based on STEP AP218”, Computer Aided Design, Vol.36, No.6, pp.537-547, 2004

20. Imagecom Inc. “2Dto3DCAD.com”, http://www.aspire3d.com

21. EMT Software Inc., “Auto-Z”, http://www.auto-z.com

22. MCS Inc., “SNAP2-3D”, http://www.mcsaz.com