

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



3D Solid Reconstruction from 2D Orthographic Views

Long Hoang

Abstract

Three-dimensional computer-aided design (CAD) models are widely used by designers because of their useful applications in the areas of CAD/CAM/CAE/CAQ. A desirous trend to create this model, which has long been studied by scientists around the world, is a 3D model reconstruction from 2D orthographic views. With this method, it is easy to enter geometric information as well as use 2D drawings that have already existed. Most of the previous works used three views, but many of the mechanical parts needed only two views. An advanced 3D solid reconstruction system using only two orthographic views is the subject of this chapter. The proposed method has been implemented and verified reliability by an ObjectARX program plugged into AutoCAD 2018. The 3D models have been checked for their compatibility with 3D CAD/CAM systems. This chapter presents principles, algorithms, databases, programming for the advanced reconstruction system, and some of its technical applications.

Keywords: 2D, 3D, reconstruction, orthographic views, drawing

1. Introduction

Currently, in the industry, there are two main types of geometric design: 2D designing shown in a multi-view drawing, which is a popular and traditional technical document, and 3D designing, which exists in the computer-aided design (CAD) and CAM systems such as Inventor, Catia, and Solidwork.

The 3D designing has many advanced applications, such as dynamic and static simulation, digital machining, visual observation, etc., and is required when we operate a CAM/CAE system. This approach has been hugely successful, initially appearing from 1990 with AutoCAD R12 and getting better and almost perfect now. Still, besides its advantages, designers should have the skills to read and understand technical drawings as well as proficiently use the 3D CAD systems, which is inconvenient for long-time engineers who are familiar with the traditional design only. Also, these 3D solid files have poor compatibility between 3D CAD software even with the same software but different versions (due to commerciality). Besides, modifying a 3D CAD file is much more complicated than editing a 2D CAD File. Additionally, the training for old engineers to get used to using 3D CAD systems instead of using 2D drawing consumes a long time. Even after having a 30-hour training course, they feel that creating two views is easier and faster than creating a 3D model, which usually uses auxiliary objects such as work plane, work axis, work point (e.g., in Inventor), etc.

With 2D designing, the designer only needs to create 2D technical drawings, which are comfortable and very familiar to the engineers. Compatibility between 2D

CAD versions is also perfect (the higher version will read the file of the lower version and can convert files of the newer version to the older version form). Besides, most of the current products have been being produced and stored by technical drawings.

Both types of design mentioned above need the CAD system can convert from one to another automatically. From 3D to 2D, the conversion process is very simple, but the reverse process (i.e., 2D to 3D that is also called reconstruction) is so complex that up to now, there is still not any software that can do it as thoroughly as we have been expecting. That is why the reconstruction problem has been studied since the first 1970s, and a large number of works can be found in the scientific literature. These can be classified into two significant categories: B-rep-oriented approach [1–14] and CSG-oriented approach [15–18].

The survey of these works allows for the following assessments: recently, the B-rep-based reconstruction approach is more appreciated than the CSG-based approach. That is mainly because CSG-based methods are less suitable for complex shapes and structures (especially when basic blocks interact, which will be difficult to identify them) and often require more user interaction than the B-rep-based method. However, there are still some limitations that exist in the B-rep-based approach as follows:

Some methods are only appropriate and proposed for polyhedral subjects. In contrast, the other authors have expanded the polyhedral approach for the objects formed by curved faces but have not yet dealt with complex intersections and interactive structures of basic blocks containing curved surfaces. Most of the reconstruction methods require the input of three views, while the technical drawings usually use only two views to describe the common machine parts. The elimination of all the invalid candidate objects is often incomplete and has not used line-type information on the views, leading to the need for more views to remove these invalid objects. There has not been a single work that has achieved all three main advantages: reconstructing a 3D solid object formed by revolving surfaces, from two views, and giving enough solutions of the 3D solid compatible with CAD/CAM systems.

This chapter presents in detail our 3D solid reconstruction system without the limitations above; that means the following have been applied:

- Using only two given views.
- Employing B-rep approach instead of the CGS.
- Extending the object domain into the solids formed by planes, cylinders, and cones.
- Outputting all solutions of the 3D solid while reducing the consumed time.
- Creating the 3D solid compatible with CAD/CAM/CAE systems.

2. Elaborating an advanced 3D solid reconstruction system

2.1 Approach

2.1.1 Typical traditional B-rep-based approach

The following synthetic reconstruction method [6] combines and develops polyhedron reconstruction methods of Wesley and Yan with Sakurai's reconstruction method of objects with curved faces. Let f be a mapping function from an object O to its view P_s , set $P_s = f(O)$. The 3D model reconstruction is to find a

reverse mapping f^{-1} such that $O^* = f^{-1}(Ps)$, where O^* is the 3D solid object model of object O . f^{-1} can be analyzed in the following five main functions:

$$f^{-1}(Ps) = fSL(fBL(fFA(fED(fVR(Ps)))))) \tag{1}$$

where fVR is the mapping function from 2D vertices in Ps to 3D vertices, fED is the mapping function from 3D vertices to 3D edges, fFA is the mapping function from 3D edges to faces, fBL is the mapping function from the faces to the candidate blocks, and fSL is the mapping function from blocks to a solid model.

In each mapping function, rules, along with some constraints, are applied to low-level objects to create higher-level objects and eliminate “ghost” elements.

Figure 1 shows the steps of a typical B-rep-based 3D model automatic reconstruction method. The method consists of eight steps. The main steps are candidate vertex formation, candidate edge formation, candidate face creation, candidate block creation, and decision-making. These steps correspond to the mapping functions in Eq. (1). When two edges intersect, they are divided into four edges in the edge segmentation step. If the two faces intersect, they are also divided into four faces in the cutting edge insertion step.

2.1.2 The author’s advanced approach

* The following are the definitions in 2D view (see **Figure 2**).

- Lines are divided into line segments by intersecting points.
- A node is an endpoint of a line segment.
- A curved line containing extreme points should be divided into two segments (e.g., a circle should be divided into two arcs).
- A view is a set of nodes and line segments.

* Definitions in 3D object (see **Figure 2**).

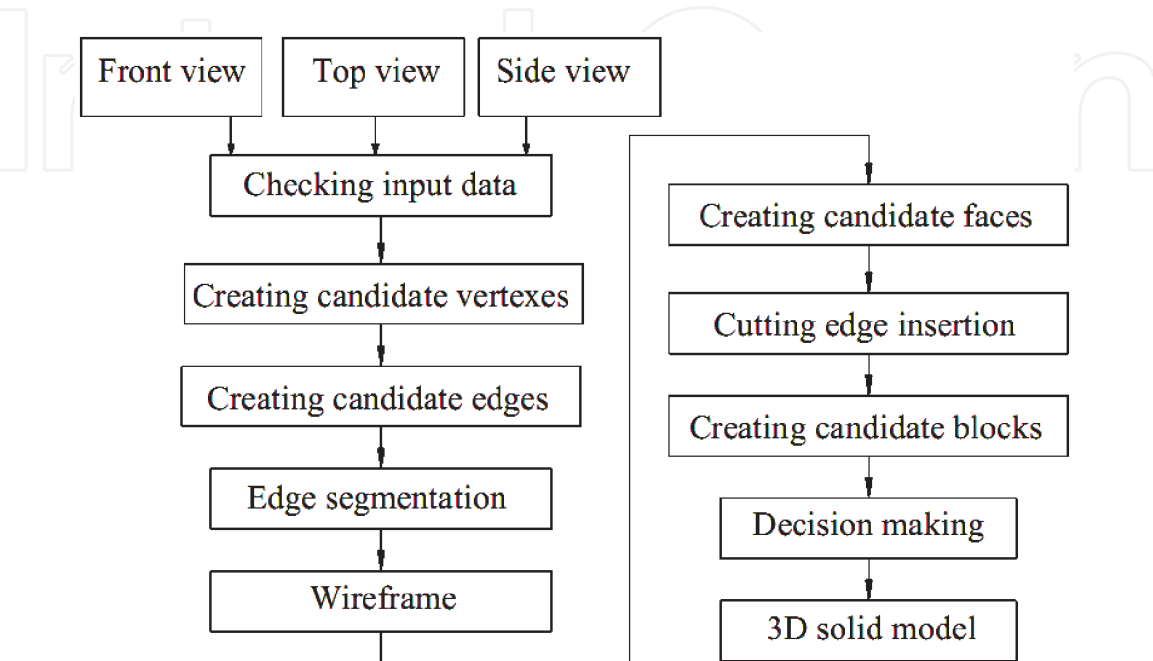


Figure 1.
Block diagram of a typical B-rep-based 3D model reconstruction method.

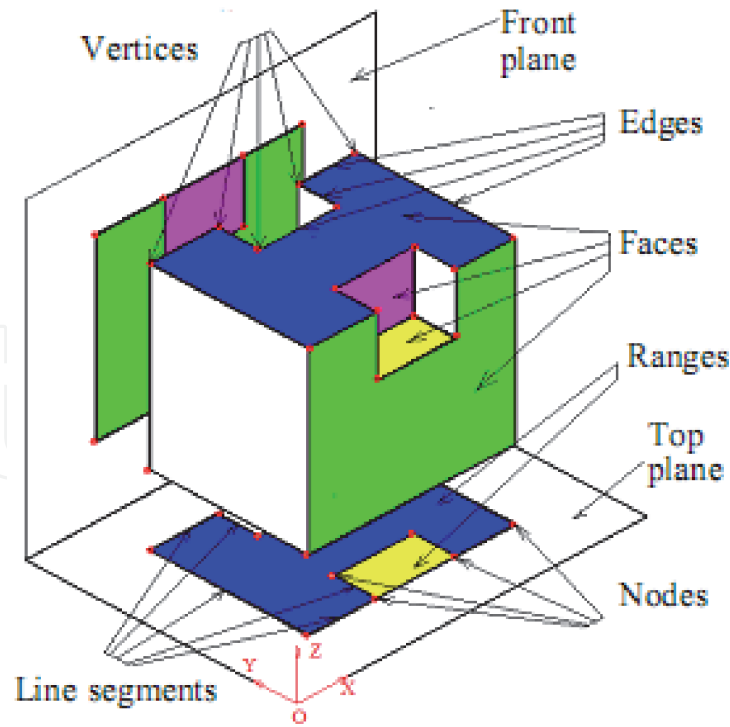


Figure 2.
Definitions in 2D view and 3D object [14].

- A solid is a body occupying a range in the three-dimensional space enclosed by several surfaces.
- A face is a segment of surface which constitutes a boundary between the solid and the exterior space.
- An edge line is an intersection of two different faces. If we desire to distinguish the line added for the identity of a curved surface from the others, the added line is called an auxiliary edge line.
- A vertex is an intersecting point of more than three edge lines.

The reconstruction problem (see **Figure 3**) is that from the front view and top view to find out the solid object O considered as a set $\{\{V\}; \{E\}; \{F\}\}$ satisfying the two groups of conditions below.

- The projection conditions:

$$O_1 \equiv \text{Front View} \quad (2)$$

$$O_2 \equiv \text{Top View} \quad (3)$$

- Topology conditions of a solid:

$$\text{An edge forms boundary of to precisely two faces} \quad (4)$$

$$\text{Two faces do not intersect at any edge except their boundary edge} \quad (5)$$

$$\text{A range is inside of projection boundaries of an even number of faces} \quad (6)$$

Where: $\{V\}$ is the set of vertices; $\{E\}$ is the set of edges; $\{F\}$ is the set of faces; O_1 is the projection of the object onto the front plane; O_2 is the projection of the

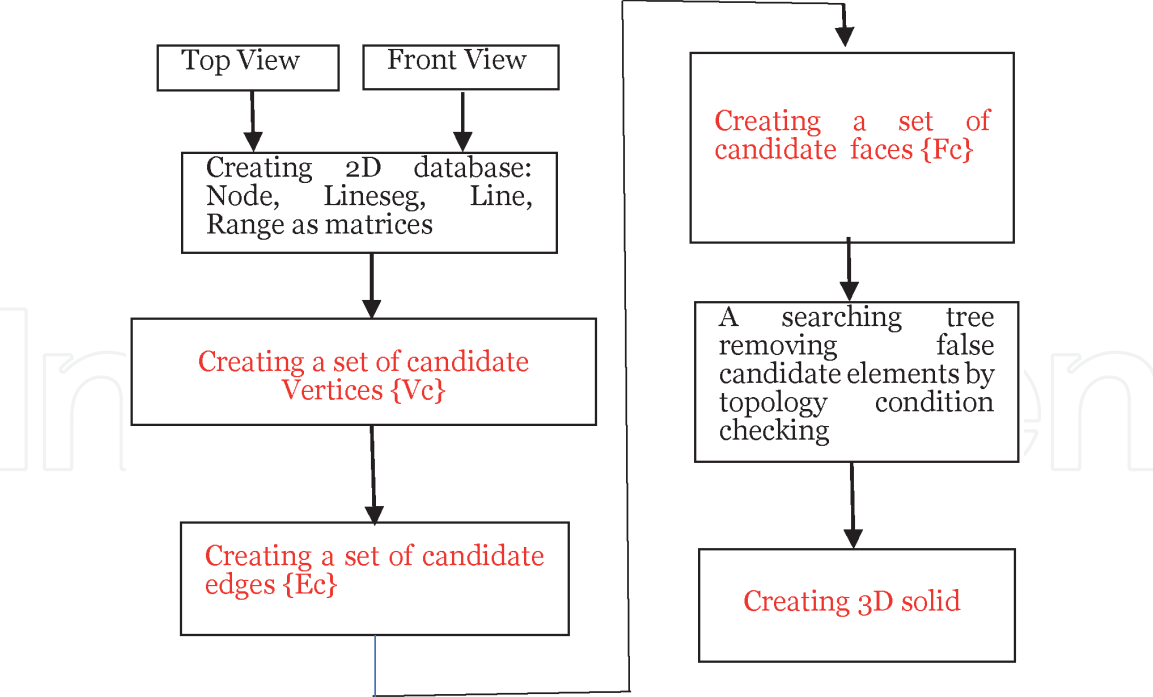


Figure 3.
Block diagram of the reconstruction system.

object onto the top plane; front view and top view are given on the 2D engineering drawing.

A general way to solve the problem consists of two main phases:

- From front view and top view, to find out a set of candidate objects (vertices, edges, faces—these objects satisfy only the condition of projection).
- To find out a subset in the set of candidate objects to meet two groups of conditions above, which means some false candidate objects must be removed.

The growing problems are:

- The algorithm to create the candidate objects should be in general for many types of surfaces such as plane, cylinder, cone, and sphere.
- The algorithm for removing false candidate elements can be against the increase in the number of the candidate. So, we need to use an efficient strategy for browsing the combination of assumed values by using the rule for the propagation of attributes (true and false) of elements, satisfying the projection and topology conditions, avoiding the combination of all.

2.2 Database and algorithms of the advanced approach

2.2.1 Specify candidate vertices

From the original database of two given views in AutoCAD that follows the DXF code, create the database as follows:

Node1[] and Node2[] are two matrices of the type ADS-POINT (used for ObjectARX programming in Microsoft Visual Studio 2015).

From database Node1[] and Node2[] above, find out any pair i, j satisfying the condition as:

$$| \text{Node1}[i][X] - \text{Node2}[j][X] | < \varepsilon \quad (7)$$

where ε is the small value depending on the user's input.

The pair i, j specifies a candidate vertex k . The algorithm for recognition of all candidate vertices is shown in **Figure 4**.

It is not difficult to specify X , Y , and Z coordinates of the 3D vertex k from its two views, which belong to the descriptive geometry as follows:

Set $Y0 = \min\{\text{Node1}[] [Y]\}$, which means we choose the lowest point on the given front view as the projection of the origin point of the coordinate system; then create the database $\text{Ver3D}[]$ that is a matrix of ADS_3D POINT:

$$-\text{ver3D}[k][Z] = \text{Node1}[i][Y] - Y0 \quad (8)$$

$$-\text{ver3D}[k][X] = \text{Node2}[j][X] \quad (9)$$

$$-\text{ver3D}[k][Y] = \text{Node2}[j][Y] \quad (10)$$

2.2.2 Specify candidate edges

From the original database of the front view, create the database as follows:

- Matrix $\text{lineseg1}[100, 2]$: $\text{lineseg1}[k][1]$ and $\text{lineseg1}[k][2]$ show two endpoints of line segment k .

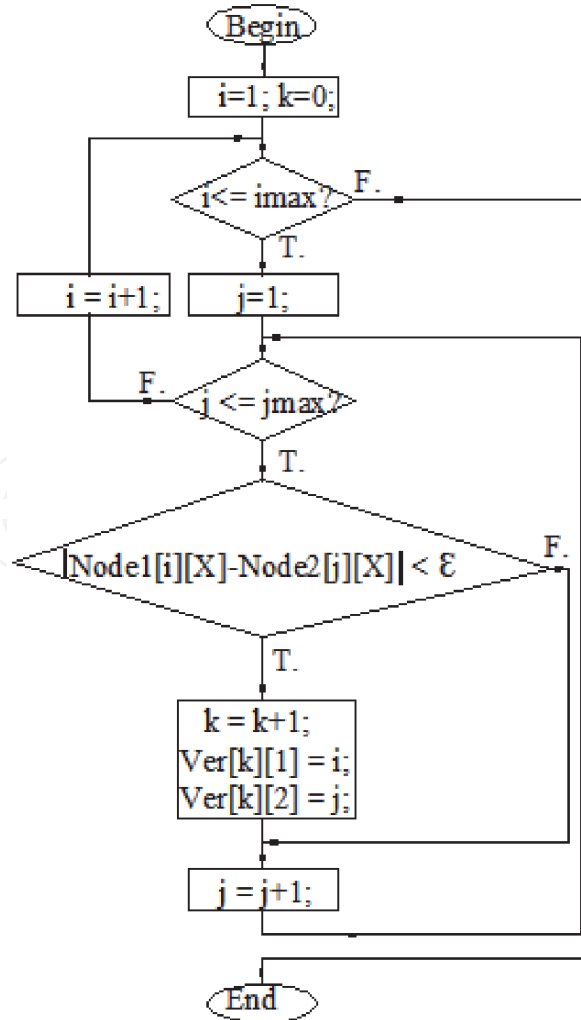


Figure 4.
The algorithm for recognition of all candidate vertices.

- Matrix $\text{line1}[100][20]$: $\text{line1}[m][i]$ contains endpoints of line segments that belong to a unique line.

It is similar to create $\text{lineseg2}[][]$ and $\text{line2}[][]$ from the top view.

From the database above, find out any pair of vertices k, m satisfying the conditions as follows:

$$\text{There is a member of } \text{line1}[][] \text{ that includes } \text{ver}[k][1] \text{ and } \text{ver}[m][1] \quad (11)$$

$$\text{There is a member of } \text{line2}[][] \text{ that includes } \text{ver}[k][2] \text{ and } \text{ver}[m][2] \quad (12)$$

The pair k, m specifies two endpoints (vertex) of a candidate edge. The set of the found edges should be stored in the matrix $\text{Ed}[100, 4]$ (100 and 4 are dimensions of the matrix).

$\text{Ed}[i][1]$ and $\text{Ed}[i][2]$ show two vertices k and m .

$\text{Ed}[i][3]$ and $\text{Ed}[i][4]$ show the members of $\text{line1}[][]$ and $\text{line2}[][]$ (i.e., $\text{Ed}[i][3]$ and $\text{Ed}[i][4]$ show front view and top view of the edge).

The conditions (11) and (12) are used for recognition of any regular edge.

If $\text{ver}[k][1] = \text{ver}[m][1]$ and they satisfy condition (12), then the pair k, m specifies two vertices of a frontal projecting edge (the edge is perpendicular to the frontal plane of projection).

If $\text{ver}[k][2] = \text{ver}[m][2]$ and they satisfy condition (11), then the pair k, m specifies two vertices of a horizontal projecting edge (the edge is perpendicular to the horizontal plane of projection).

Note: for each edge, check for the possible existence of intermediate vertices. If an intermediate vertex is found, the vertex causes the creation of two new edges (unless one of them already exists).

2.2.3 Specify candidate faces

- Projecting face (the face is perpendicular to the plane of projection)

For each member j of the matrix $\text{line1}[][]$, find out all of the edges i as follows:

$$\text{fronted} = \text{ed}[i][3]; \text{begin} = \text{ed}[i][1]; \text{end} = \text{ed}[i][2]; \text{frontbegin} = \text{ver}[\text{begin}][1];$$

$$\text{frontend} = \text{ver}[\text{end}][1].$$

If frontbegin is equal to frontend and they belong to $\text{line1}[j][i]$, then the edge i is a frontal projecting edge that belongs to the surface j (that has front view as the line corresponding to $\text{line1}[j][i]$). (13)

If frontbegin is not equal to frontend but fronted is equal to j , then edge i belongs to the surface j (14)

From the set of edges on the frontal projecting surface j , find out all minimal closed loops of edges, they specify a new candidate face. The algorithm for recognition of frontal projecting faces is shown in **Figure 5**. It is similar to recognizing horizontal projecting faces. The candidate faces should be stored in the database as follows:

$\text{Faceed}[100][30]$: $\text{faceed}[k][0]$ shows the number of edges that belong to the face k ; $\text{faceed}[k][i]$ shows edge i of face k , $\text{faceed}[k][29]$ equal to j mentioned above that specify geometry information of the face k (100 and 30 are dimensions of the matrix).

- ii. Cylinder. The cylinders mentioned here are projecting cylinders so that in any view, one projection of the cylinder becomes a circle. The circle should be divided into two arcs. The cylinder is divided into two half projecting cylinders; the algorithm for recognition of these half cylinders is the same as the algorithm above.
- iii. Cone. The axis of the cone mentioned here is perpendicular to the plane of projections so that in any view, one projection of the cone becomes two circles (one of them may be a point). The two circles were divided into four arcs, which mean the cone is divided into two half cones.

2.2.4 Removing false elements

A searching tree removes false candidate elements by checking conditions (5), (6), and (7). The purpose of this traversal process is to eliminate false assumptions due to dissatisfaction with topological conditions. To counteract the increase in browsing time on assumption binary tree, the duration of this process is exponentially increased: 2^n , where n is the number of assumed faces. Status management and browse planning aim at selecting the face for the next step with the highest priority (the priority is assessed by its amount of information, for example, the face

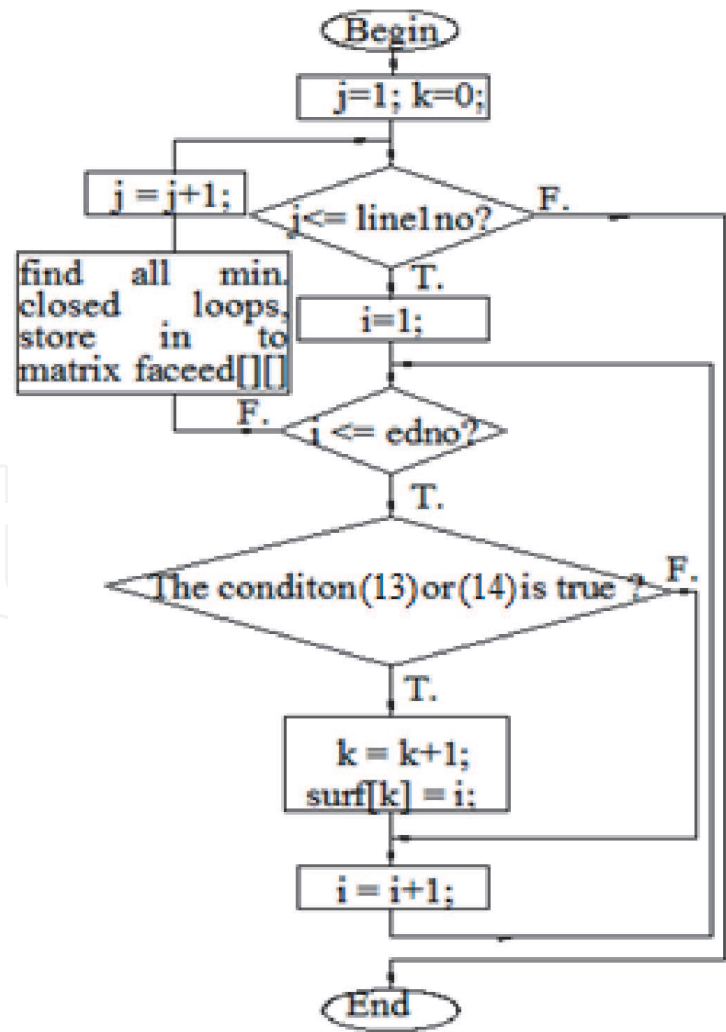


Figure 5.
Algorithm for recognition of projecting faces.

contains many edges, and the face will see a high level of priority). Conflicts are found during the browsing process. If it meets any conflict, the next step will be backtracked.

2.2.5 Solid creation

Based on the condition (5), the algorithm to create the solid is described as follows: for each range, the faces are numbered in height order; the primitive solid is generated from first to second, third to fourth, etc. The outcome solid is a union of all such primitive solids (see **Figure 7b**).

2.3 Implementation and verification results

The proposed reconstruction method has verified reliability by a program written in Visual Studio (see **Figure 6**). The program was compiled and then built an Objectarx file to run in AutoCAD. After downloading the Objectarx file, AutoCAD has an extended command to rebuild the 3D solid model from its two views (see **Figure 7**). The 3D solid model has been exported as the SAT file that PTC Creo Parametric 3.0 (CAM software package) can use. The tool paths generated in PTC Creo (see **Figure 7c**) have been compiled into the specific codes needed for the HS Super MC500 CNC machine to mill the surfaces. The machined part was then 3D scanned. The 3D comparison result generated by Geomagic software is shown in **Figure 7(d)**. The machining accuracy in **Figure 7(d)** indicates that the 3D solid model reconstructed from its two views is compatible and usable for CAD/CAM/CAQ/CNC systems. The proposed method is limited to perfect input drawings that contain only lines, circles, and arcs. However, an engineering drawing is often a mixture of geometric representations and annotations, and it's challenging to ensure that engineering drawings are absolutely accurate. Therefore, techniques to reconstruct the 3D solid model from real drawings should consider these imperfections [6].

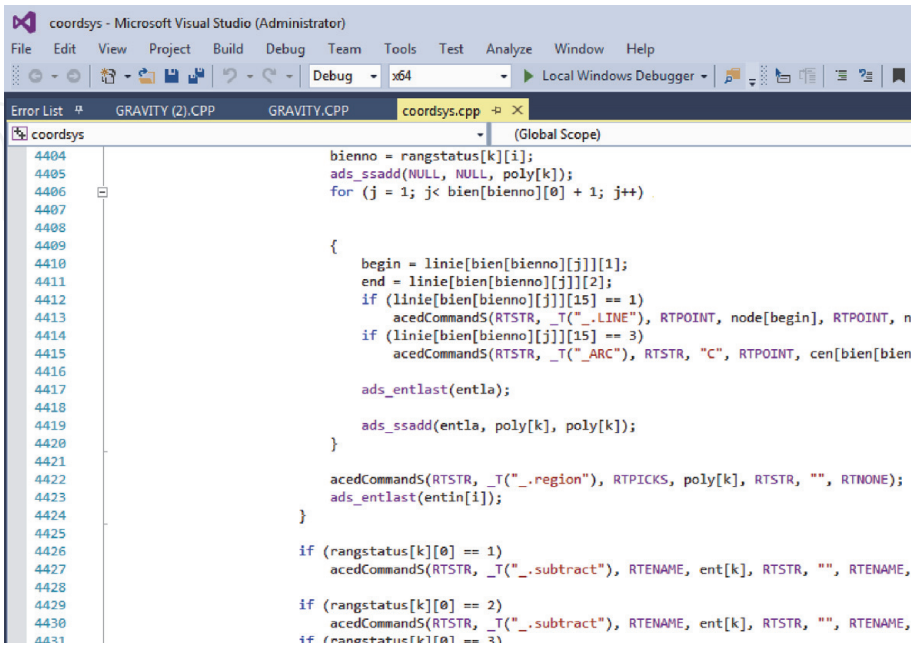


Figure 6.
Extracting of ObjectARX program in Microsoft visual studio 2015.

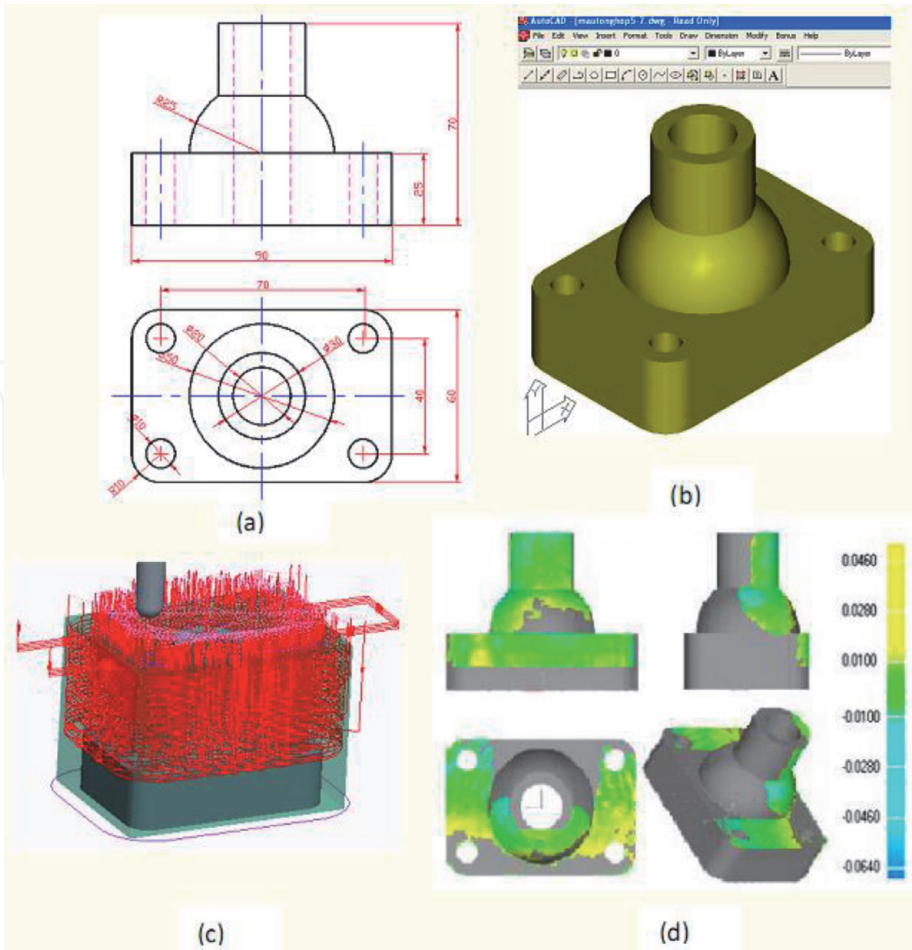


Figure 7.
(a) Two given views; (b) solid creating automatically; (c) tool path generation; (d) 3D comparison between the CAD model and the part machined by CNC.

3. Conclusions

The 3D solid models are extremely useful in techniques. An excellent way to create the 3D solid is an automatic reconstruction from its views. The 3D solid automatic reconstruction system presented in this chapter has advanced features as follows:

- Using only two given views.
- Outputting all solutions of the 3D solid while reducing the consumed time.
- Creating the 3D solid compatible with CAD/CAM/CAE systems.

IntechOpen

IntechOpen

Author details

Long Hoang
Ha Noi University of Science and Technology, Vietnam

*Address all correspondence to: long.hoang@mail.hust.edu.vn

IntechOpen

© 2020 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Idesawa M. A system to generate a solid figure from three view. *Bulletin of JSME*. 1973;**16**(92):216-225
- [2] Wesley MA, Markowsky G. Fleshing out projections. *IBM Journal of Research and Development*. 1981;**25**(6):229-258
- [3] Sakurai H, Gossard DC. Solid model input through orthographic views. *ACM SIGGRAPH Computer Graphics*. 1983; **17**(3):243-252
- [4] Dutta D, Srinivas YL. Reconstructing curved solids from two polygonal orthographic views. *Computer-Aided Design*. 1992;**24**(3):149-159
- [5] You CF, Yang SS. Automatic feature recognition from engineering drawings. *The International Journal of Advanced Manufacturing Technology*. 1998;**14**(7): 495-507
- [6] Watanabe T. Revision of inconsistent orthographic views. *Journal for Geometry and Graphics*. 1998;**2**(1): 45-53
- [7] Shin BS, Shin YG. Fast 3D solid model reconstruction from orthographic views. *Computer-Aided Design*. 1998; **30**(1):63-76
- [8] Liu SX et al. Reconstruction of curved solids from engineering drawings. *Computer-Aided Design*. 2001;**33**(14):1059-1072
- [9] Furferi R, Governi L, Palai M, Volpe. 3D model retrieval from mechanical drawings analysis. *International Journal of Mechanics*. 2011:91-99
- [10] Long H, Long BT. Automatic creating 3D pseudo-wireframe from 2D orthographic views. *Journal of Science and Technology of Ha Noi University of Science and Technology*. 2015;**106**: 46-49
- [11] Long H, Long BT, Van Hieu P. Conical solid model reconstruction of 3D pseudo-wireframe model found from 2D orthographic views. *Journal of Science and Technology of Ha Noi University of Science and Technology*. 2015;**108**:68-72
- [12] Long H, Long BT. Automatic 3D model reconstruction from a multi-views engineering drawing file containing even curves and hidden lines for CAD/CAM systems. In: *Proceedings RCMME*. 2014. pp. 20-23. ISBN: 978-604-911-942-2
- [13] Long H. Expanding a 3D solid reconstruction system using two views to the system using three views. *Journal of Science and Technology of Ha Noi University of science and technology*. 2018;**125**:40-45
- [14] Hoang L, Tien LB. A flexible solid 3D model reconstruction system for mechanical CAD/CAM systems. *Journal of the Korean Society for Precision Engineering*. 2019;**36**(8): 753-759
- [15] Aldefeld B. On automatic recognition of 3-D structures from 2-D representations. *Computer-Aided Design*. 1983;**15**:59-64
- [16] Geng W, Wang J, Zhang Y. Embedding visual cognition in 3D reconstruction from multi-view engineering drawings. *Computer-Aided Design*. 2002;**34**(4):321-336
- [17] Lee H, Han S. Reconstruction of 3D interacting solids of revolution from 2D orthographic views. *Computer-Aided Design*. 2005;**37**(13):1388-1398
- [18] Wang Z, Latif M. Reconstruction of 3D solid models using fuzzy logic recognition. *Proceedings of the World Congress on Engineering*. 2007;**1**:37-42