Developing an Automatic 3D Solid Reconstruction System from only Two 2D Views

Long Hoang
School of Mechanical Engineering, Hanoi University of Science and Technology, Vietnam
long.hoang@hust.edu.vn (corresponding author)

Thanh Tuan Nguyen
School of Mechanical Engineering, Hanoi University of Science and Technology, Vietnam
tuan.nguyenthanh@hust.edu.vn

Hoang Anh Tran
School of Mechanical Engineering, Hanoi University of Science and Technology, Vietnam
anh.tranhoang@hust.edu.vn

Duc Huy Nguyen
School of Mechanical Engineering, Hanoi University of Science and Technology, Vietnam
huy.nguyenduc@hust.edu.vn

Received: 14 June 2024 | Revised: 28 June 2024 | Accepted: 30 June 2024

 DOI: https://doi.org/10.48084/etasr.8141

ABSTRACT

Three-dimensional (3D) solid models of mechanical machine parts are widely used in modern mechanical engineering. One expected approach in creating 3D models is automatically reconstructing them from 2D engineering drawings. This work expands the previous automatic reconstruction methods by adding principles, algorithms, and coding to reconstruct oblique planes on the part. The proposed method uses only two views as the input to reconstruct the 3D solid part, including oblique planes. The proposed method has been implemented by a program written in the ARX 2018 language running on the AutoCAD 2021 platform to reconstruct multiple 3D parts from their two views. Experimental test results on many samples confirmed that the proposed method is reliable, absolutely accurate, and achieves a high reconstruction speed. The output 3D model has also been tested and confirmed for compatibility with CAD/CAM software such as Solid Works, Inventor, and PTC Creo.

Keywords-3D solid model; CAD; CAM; orthographic views; 3D reconstruction

I. INTRODUCTION

Three-dimensional (3D) solid models of a machine part are very useful in modern mechanical engineering, including various fields such as stress and displacement analysis [1] and optimizing the part structure to improve the performance and quality of 3D printed products [2]. However, traditional 2D technical drawings are still popularly circulated as official documents in design. In addition, using 2D CAD systems to create technical drawings is also easy and friendly for many engineers. These lead to a need for automated reconstruction from 2D view drawing to a 3D solid model for use in modern mechanical engineering. The above reconstruction problem is so difficult that no commercial software can perform it well, although many scientists have researched it for a long time, starting from Idesawa [3] with his study on automatic reconstruction for polyhedra using three views. Then, a series of other authors have proposed methods for the reconstruction of more diverse surfaces; they can be divided according to the face-oriented approach (also called Boundary Representation - BREP) [3-11, 15-17] or volume-oriented approach (also called Constructive Solid Geometry - CGS) [12-14], and each work is limited to the types of surfaces and their intersection curves. Most previous works use three views as the input [3, 4, 6-14], while many technical drawings only contain two views. On other sites, authors in [5, 15-17] have proposed an advanced method which uses only two views as the input to reconstruct the 3D part, but they have not mentioned the oblique plane on the reconstructed part. The problem of automatically reconstructing 3D models from its views is very difficult. Each work has contributed to expanding the scope of objects, reducing the number of given views, expanding the input type, correcting drawing errors, and handling drawings that lack hidden lines.

Due to the limitations of previous works mentioned above, this work expands the two views-based reconstruction method in [15-17], shown in brief below, by adding principle, algorithm, and coding to reconstruct oblique planes on the part, which will be presented in detail.

II. THE BASIC TWO VIEWS-BASED RECONSTRUCTION METHOD

From two given views on the engineering drawing file that can be imported into the AutoCAD package, the reconstruction problem is finding the 3D part specified by a set of vertexes{V}, edges {E}, and faces {F} [15] which satisfies the projection conditions and topology conditions below:

Frontal project{{V}, {E}, {F}} ≡

The given Front View (1)

Horizontal project{{V}, {E}, {F}} ≡

The given Front View (2)

Each edge is connected to precisely two faces (3)

Two faces intersect only at their edges (4)

Each region is the projection of

an even number of faces (5)

The vertexes, edges, and faces of the reconstruction system satisfying only the conditions (1) and (2) are called candidate vertexes, candidate edges, and candidate faces, respectively [15]. The candidates may be false or true, which is checked by conditions (3), (4), (5). Algorithms to find candidate vertexes, candidate edges, and candidate faces, including projecting planes, projecting cylinders, cones, and spheres, have been presented in detail [15-17], but oblique planes have not been presented, which will be presented in the next section below.

III. EXPANDING: PRINCIPLE, ALGORITHM, AND CODING FOR OBLIQUE PLANES

A. Principle

The equation of a plane is specified by three points that do not belong to a straight line: A(X_{A}, Y_{A}, Z_{A}), B(X_{B}, Y_{B}, Z_{B}), and C(X_C, Y_C, Z_{C}), as follows:

The normal vector of the plane is specified as:

$$\vec{N} = \left(\overrightarrow{AB}.\overrightarrow{AC}\right) = (a, b, c) \quad (6)$$

The equation of the plane is specified by point A(X_{A}, Y_{A}, Z_{A}), and the normal vector $\vec{N}(a, b, c)$, is:

$$a(X\text{-}X_A) + b(Y\text{-}Y_A) + c(Z\text{-}Z_A) = 0 \quad (7)$$

From the three given points, A, B, and C, the normal vector $\vec{N}$ is calculated by (6), and the condition for any point (X, Y, Z) on the plane is checked based by (7).

B. Algorithm

The algorithm for finding candidate oblique planes is described by the following pseudocode:

```
The set of candidate oblique planes is set
to empty;
Repeat ( number of candidate vertexes)
{Repeat (number of edges pair connected to
the selected vertex)
{
Specify three different endpoints of two
selected edges;
Repeat ( number of planes in the candidate
oblique plane set)
{
If the plane specified by the three
endpoints is not the same as all of the
selected planes in the candidate oblique
plane set, add a new plane to the
candidate oblique plane set;
}
}
```

The algorithm for finding candidate edges on candidate oblique planes is described by the following pseudocode:

```
Repeat ( number of planes in the candidate
oblique plane set)
{
Specify three vertexes stored in the
selected plane;
Repeat ( number of candidate edges)
{
Specify two endpoints of the selected
edge;
If both endpoints lie on the selected
plane specified by the three vertexes,
then add the edge to the selected plane;
}
}
```

C. Coding

An extract of the ARX codes for finding candidate edges on candidate oblique planes follows.

```
For (ii = 1; ii <= mpbkno; ii++) // Repeat
for the number of planes in the candidate
oblique plane set.
{ sl = 0; // counter the number of edges
// Specify three points on the plane
dinh1 = mpbkfunt[ii][2];
dinh2 = mpbkfunt[ii][3];
dinh3 = mpbkfunt[ii][4];
delX21 = ver3d[dinh2][X] -
ver3d[dinh1][X];
delY21 = ver3d[dinh2][Y] -
ver3d[dinh1][Y];
delZ21 = ver3d[dinh2][Z] -
ver3d[dinh1][Z];
```

```
delX31 = ver3d[dinh3][X] -
ver3d[dinh1][X];
delY31 = ver3d[dinh3][Y] -
ver3d[dinh1][Y];
delZ31 = ver3d[dinh3][Z] -
ver3d[dinh1][Z];
// Specify the normal vector of the plane
ptX = delY21 * delZ31 - delY31 * delZ21;
ptY = delZ21 * delX31 - delZ31 * delX21;
ptZ = delX21 * delY31 - delX31 * delY21;
for (j = 0; j < edgenum; j++)
                  facetem[j] = 0;
for (j = 0; j < edgenum; j++) // Repeat
for number of candidate edges.
{ // Specify two endpoints of the edge
dinh1l = ed[j][0];
dinh2l = ed[j][1];
dinh1lX = ver3d[dinh1l][X];
dinh1lY = ver3d[dinh1l][Y];
dinh1lZ = ver3d[dinh1l][Z];
dinh2lX = ver3d[dinh2l][X];
dinh2lY = ver3d[dinh2l][Y];
dinh2lZ = ver3d[dinh2l][Z];
dinh1X = ver3d[dinh1][X];
dinh1Y = ver3d[dinh1][Y];
dinh1Z = ver3d[dinh1][Z];
delX1 = dinh1lX - dinh1X;
delY1 = dinh1lY - dinh1Y;
delZ1 = dinh1lZ - dinh1Z;
delX2 = dinh2lX - dinh1X;
delY2 = dinh2lY - dinh1Y;
delZ2 = dinh2lZ - dinh1Z;
if ((abs(ptX * delX1 + ptY * delY1 + ptZ *
delZ1)<0.3) && (abs(ptX * delX2 + ptY *
delY2 + ptZ * delZ2) < 0.3))
{ sl++; // Checking condition (7)
facetem[sl] = j;  }} //adding the edge  j
to the selected plane.
```

IV. TESTING RESULTS AND DISCUSSION

The basic reconstruction method presented in section II and its expansion presented in section III has been implemented by a program written in ARX 18 language running on AutoCAD 2021 platform to reconstruct for part 1, as shown in Figure 2, which is given by two views in Figure 1. The final 3D model was exported to the N.X. 12 platform to check its accuracy and compatibility with 3D CAD systems, as shown in Figure 3.

Part 2 is given by two views (Figure 4) and is reconstructed by the program. There are two solutions, and the final 3D solids are shown in Figure 5 and Figure 6.

With the methods in [15-17] presented in Section II, the number of solutions for the part given by the two views in Figure 4 is only one, as shown in Figure 5, because they find only projecting planes. Expanding the method by adding oblique planes, as presented in section III, the program has found two solutions. The first is shown in Figure 5, and the second is shown in Figure 6.

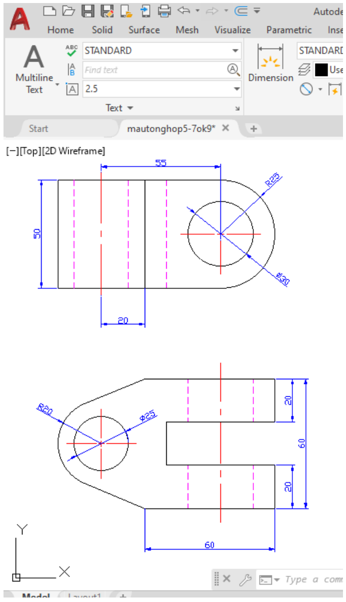

Fig. 1. Two given views of part 1.

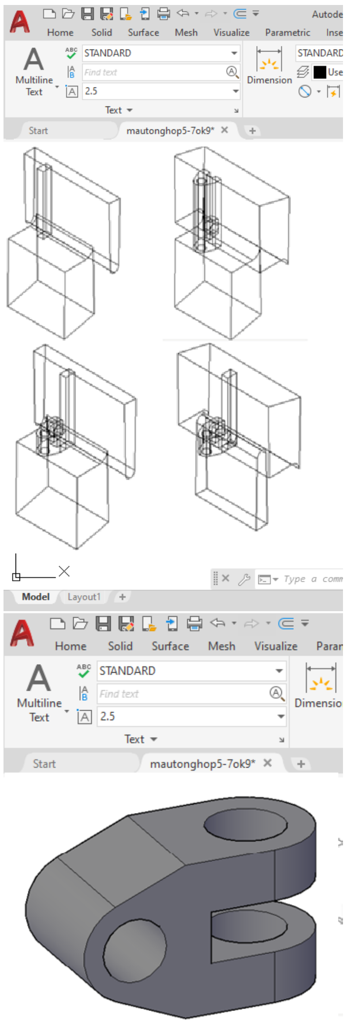

Fig. 2. Automatic reconstruction process and the final 3D solid.

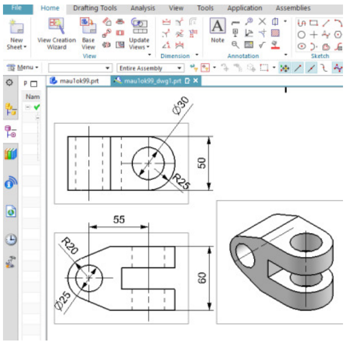

Fig. 3. Absolutely compatible with the N.X. 12 platform.

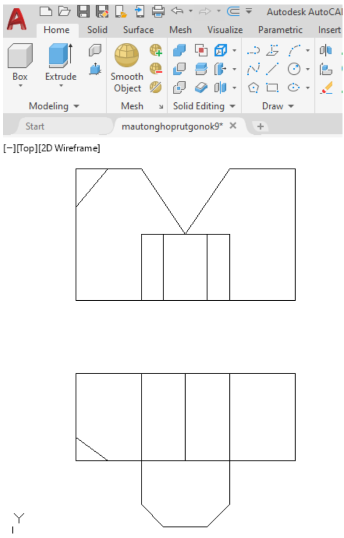

Fig. 4. Two given Views of part 2.

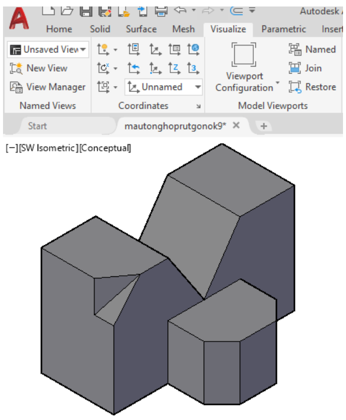

Fig. 5. The first solution of the final 3D solid.

Based on the proposed method, the reconstruction process consumes a little more time than the methods that use three views because the number of assum faces is greater. Still, given a today's modest computer (Intel core i5 CPU 2.6 GHz, RAM 16 G.B.), the consuming time is just under 3 seconds, while the accuracy of the reconstructed 3D models of all methods is the same, as shown in Figure 3, but the outstanding advantage of the proposed method is that it requires only two views compared to the methods based on three views of most previous works [3, 4, 6-14]. If there are more solutions of the 3D reconstructed solid, the program automatically creates all of them, so the designer can choose one. This is easier and faster than drawing three views. Of course, the mechanical drawing must contain enough views, even including sectional views, to find only one solution. However, when creating a 3D solid, we should use or give at least two views.

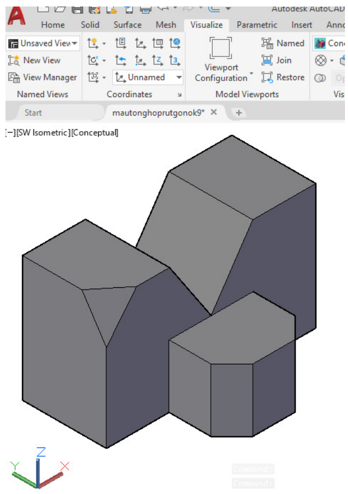

Fig. 6. The second solution of the final 3D solid.

V. CONCLUSION

This paper presents a two-view-based 3D reconstruction method, including new contributions that expand the scope of reconstructed parts. The proposed method has the following advantages:

- Use the least views for reconstruction
- Finds out all of the solutions for the reconstructed part, including even oblique planes.
- The reconstructed 3D part is absolutely accurate.
- The 3D solid model of the reconstructed part is compatible with most 3D CAD platforms, which allows the designer to use the reconstructed 3D solid model in CAD/CAM/CAE/CNC.

ACKNOWLEDGMENT

The paper expands on the research done within the project B2014-218, which the Vietnam Ministry of Education has supported.

REFERENCES

[1] F. A. Alnaimat, "The Geometrical Effect on the Von Mises Stress on Ball and Socket Artificial Discs," *Engineering, Technology & Applied Science Research*, vol. 10, no. 5, pp. 6330–6334, Oct. 2020, https://doi.org/10.48084/etasr.3789.

[2] N. T. Anh, N. X. Quynh, and T. T. Tung, "Study on Topology Optimization Design for Additive Manufacturing," *Engineering, Technology & Applied Science Research*, vol. 14, no. 3, pp. 14437–14441, Jun. 2024, https://doi.org/10.48084/etasr.7220.

[3] M. Idesawa, "A System to Generate a Solid Figure from Three View," *Bulletin of JSME*, vol. 16, no. 92, pp. 216–225, 1973, https://doi.org/10.1299/jsme1958.16.216.

[4] M. A. Wesley and G. Markowsky, "Fleshing Out Projections," *IBM Journal of Research and Development*, vol. 25, no. 6, pp. 934–954, Nov. 1981, https://doi.org/10.1147/rd.256.0934.

[5] H. Sakurai and D. C. Gossard, "Solid model input through orthographic views," *ACM SIGGRAPH Computer Graphics*, vol. 17, no. 3, pp. 243–252, Apr. 1983, https://doi.org/10.1145/964967.801155.

[6] C. F. You and S. S. Yang, "Automatic feature recognition from engineering drawings," *The International Journal of Advanced Manufacturing Technology*, vol. 14, no. 7, pp. 495–507, Jul. 1998, https://doi.org/10.1007/BF01351395.

[7] T. Watanabe, "Revision of Inconsistent Orthographic Views," *Journal for Geometry and Graphics*, vol. 2, no. 1, pp. 45–53, 1998.

[8] B.-S. Shin and Y. G. Shin, "Fast 3D solid model reconstruction from orthographic views," *Computer-Aided Design*, vol. 30, no. 1, pp. 63–76, Jan. 1998, https://doi.org/10.1016/S0010-4485(97)00054-7.

[9] S.-X. Liu, S.-M. Hu, Y.-J. Chen, and J.-G. Sun, "Reconstruction of curved solids from engineering drawings," *Computer-Aided Design*, vol. 33, no. 14, pp. 1059–1072, Dec. 2001, https://doi.org/10.1016/S0010-4485(00)00143-3.

[10] R. Furferi, L. Governi, Y. Volpe, and M. Palai, "3D model retrieval from mechanical drawings analysis," *International Journal of Mechanics*, vol. 5, no. 2, pp. 91–99, 2011.

[11] W. Geng, J. Wang, and Y. Zhang, "Embedding visual cognition in 3D reconstruction from multi-view engineering drawings," *Computer-Aided Design*, vol. 34, no. 4, pp. 321–336, Apr. 2002, https://doi.org/10.1016/S0010-4485(01)00092-6.

[12] H. Lee and S. Han, "Reconstruction of 3D interacting solids of revolution from 2D orthographic views," *Computer-Aided Design*, vol. 37, no. 13, pp. 1388–1398, Nov. 2005, https://doi.org/10.1016/j.cad.2005.01.007.

[13] Z. Wang and M. Latif, "Reconstruction of 3D Solid Models Using Fuzzy Logic Recognition," in *World Congress on Engineering*, London, UK, Jul. 2007, pp. 1–6.

[14] B. Aldefeld, "On automatic recognition of 3D structures from 2D representations," *Computer-Aided Design*, vol. 15, no. 2, pp. 59–64, Mar. 1983, https://doi.org/10.1016/0010-4485(83)90169-0.

[15] L. Hoang, "A 3D solid model reconstruction system from only two views for CAD/CAM/CAE/CNC," *International Journal of Modern Physics B*, vol. 34, no. 22n24, Sep. 2020, Art. no. 2040157, https://doi.org/10.1142/S0217979220401578.

[16] L. Hoang and L. B. Tien, "A Flexible Solid 3D Model Reconstruction System for Mechanical CAD / CAM Systems," *Journal of the Korean Society for Precision Engineering*, vol. 36, no. 8, pp. 753–759, Aug. 2019, https://doi.org/10.7736/KSPE.2019.36.8.753.

[17] L. Hoang, "Automatic Creating 3D Solid Models for Advanced Mechanical Engineering Applications," in *International Conference on Advanced Mechanical Engineering, Automation, and Sustainable Development*, Ha Long, Vietnam, Nov. 2021, pp. 468–473, https://doi.org/10.1007/978-3-030-99666-6_68.