www.cadanda.com Computer-Aided Design AND APPLICATIONS

Method to Automatically Convert Sketches of Mechanical Objects into 3D Models

Masaji Tanaka[1], Motoki Terano[1], Tetsuya Asano[2] and Chiharu Higashino[2]

[1]Okayama University of Science, {tanaka, m_terano}@mech.ous.ac.jp
[2]Aikoku Alpha Corporation, {t-asano, higashino}@aikoku.com

Corresponding author: Masaji Tanaka, tanaka@mech.ous.ac.jp

Abstract. In a computerized society, the importance of three-dimensional (3D) models is increasing rapidly. Although solid modelers are popular tools, sketches such as two-dimensional (2D) line drawings are important, especially for designers, because sketches of 3D objects can be drawn easily. We have developed a method to automatically convert sketches into 3D models. However, convertible sketches are strictly limited in the method. Herein, a new method is proposed that can handle more sketches compared with our past method. In this method, sketch faces are introduced and various types of sketch features are defined to obtain solutions. Herein, a detailed algorithm is explained, and four examples are indicated to clarify the effectiveness and practicability of this method.

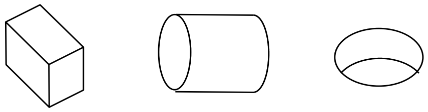
Keywords: automatic conversion, sketch, 3D model, sketch face, sketch feature.
DOI: https://doi.org/10.14733/cadaps.2020.1168-1176

1 INTRODUCTION

Presently solid modelers have become popular tools in computer-aided design/manufacturing (CAD/CAM) and computer graphics (CG) systems. However, for designers, primary ideas of creative objects would typically be drawn in sketches of line drawings because creating solid models from the ideas directly is time consuming and difficult. Furthermore, to express 3D objects, it is easy to draw sketches rather than using solid modelers. A system that can automatically convert a sketch into a 3D model as a solid model would be an effective tool for conveying the ideas for designers and others.

Many methods have been proposed for the conversion of sketches into 3D models, e.g. [4]. The Clowes--Huffman line labelling technique was developed [3],[10], where sketches do not contain any hidden line and each sketch is drawn precisely from a general view of a 3D object. In the technique, when a sketch as a line drawing is input, each line segment can be labeled as "+", "-" or arrow, and several junction types are applied, such as Y-junction, W-junction, and T-junction. Each junction can express a convex or a concave shape of a 3D object drawn in the sketch. For example, if a Y-junction consists of three "+" lines, it expresses a convex shape. Moreover, Varley et al. [17] demonstrated how 3D polyhedrons such as 3D cuboids were created from their sketches mathematically. Although the technique was effective for recognizing 3D objects from sketches, the sketches could not include curved lines. Therefore, only sketches of polyhedrons could be addressed. Malik [12] extended the technique to curved lines. However, he did not indicate how 3D models could be generated from sketches including curved lines. Additionally, the models were addressed from various aspects, e.g. [5-6],[13], although their targets were strictly fixed. Lee and Han [11] proposed a method to reconstruct solids of revolution from mechanical drawings. Although the method was effective for only rotational objects, the idea to recognize their 2D drawings was similar to our method. Meanwhile, several computer-aided 3D sketching (CAS) systems have been developed, e.g. [2],[8-9]. It appears that these systems attempt to develop easier solid modelers for beginners. However, they are not ordinal solid modelers such as CATIA and SolidWorks. Therefore, their preciseness may be worse than those of solid modelers and their usefulness may be worse than those of full automatic conversion systems such as our system described below.

We have developed a method for the automatic conversion of sketches including curved lines into 3D objects since approximately five years ago [15-16]. In the method, sketch features were first defined. Figure 1 shows three samples of them. Most humans can detect each sketch feature as a 3D model. Therefore, if a complex sketch consists of many sketch features, the sketch can be converted into a 3D model by detecting and extracting each sketch feature step by step. In this case, sketch features can include curved lines such as cylinders and round holes. Therefore, sketches including curved lines can be addressed easily. The algorithm of the method was formed by the repetition of the detection and the extraction of sketch features. However, when a sketch feature was extracted from a sketch, isolated lines each of which do not form any loops of lines appeared frequently. They prevented the next detection of a sketch feature. Therefore, we applied a machine-learning technique for the restoration of the isolated lines. Consequently, we found it was not impossible that the method converts sketches into 3D models although their shapes were limited. We have been continuously developing a detailed algorithm of the learning system because of the implementation of the method. The next target of our research is to extend convertible sketches with the method. We herein propose a new method that can address significantly more sketches than our past method although the 3D objects of the sketches are limited to mechanical objects.



(a) Cuboid (b) Cylinder (c) Hole

Figure 1: Three samples of sketch features.

In this study, we first define various types of sketch faces that can form sketch features. Next, the precise algorithm of this method is indicated. In the algorithm, the introduction of our learning technique would be minimized because machine-learning systems are generally effective for ambiguous problems; however, correct solutions are not always obtained. Therefore, the computational time to convert a sketch into a 3D model would be minimized in this method. Furthermore, we consider the ambiguity to recognize sketches with hidden lines. For example, T-junctions tend to produce complex hidden lines, e.g. [1]. In this method, we apply the characters and tendencies of mechanical objects for this problem. Therefore, more natural solutions such as 3D models would be obtained from the sketches. The algorithm of this method has been verified partially with experimental systems implemented in Visual C++.

2 SKETCH FACES AND SKETCH FEATURES

As shown in Figure 1, a cuboid sketch consists of three parallelograms, a cylinder sketch consists of an ellipse and a cylindrical face, and a hole sketch comprises an ellipse and an elliptical arc. Generally, humans believe that a parallelogram corresponds to a 3D rectangle and an ellipse corresponds to a 3D circle in the sketches. These faces are called sketch faces in this method. To address more types of sketch features, they are defined from sketch faces in this method. Seven sketch faces in Figure 2 and nine sketch features in Figure 3 are addressed in this method. For example, a curved parallelogram consists of two straight lines and two elliptical arcs or Bézier curves. Furthermore, a cylinder sketch comprises an ellipse and a curved parallelogram, a concave triangular prism comprises a triangle and a parallelogram, a prism sketch comprises a polygon and parallelograms, and an extrusion sketch comprises a complex face and parallelogram(s) and/or curved parallelogram(s). Sketches created from these sketch features resemble mechanical objects, especially machined objects; furthermore, because they are familiar to designers, only they are addressed in this method.

For a clearer explanation, this method addresses only precise sketches of mechanical objects drawn in 2D CAD systems, and each sketch is a general view of a 3D object. It is an issue of this method to develop an automatic system that can convert rough sketches into precise ones. This issue is discussed in Section 5. Furthermore, each sketch feature can correspond to a 3D model in this method. This conversion is based on [17]. Quantitative analyses such as the accuracy and scalability of the 3D model will be addressed next because we wish to demonstrate the possibility of developing a full automatic conversion system from various types of sketches to 3D models in this study.

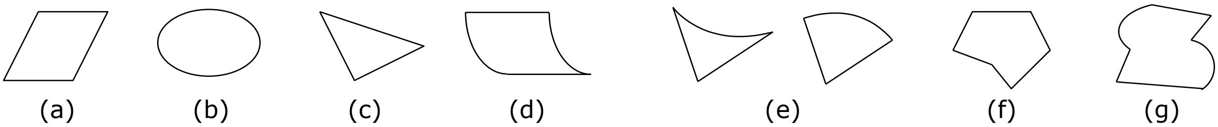


Figure 2: Seven sketch faces: (a) Parallelogram, (b) Ellipse, (c) Triangle, (d) Curved parallelogram, (e) Two types of curved triangle, (f) Polygon, and (g) Complex.

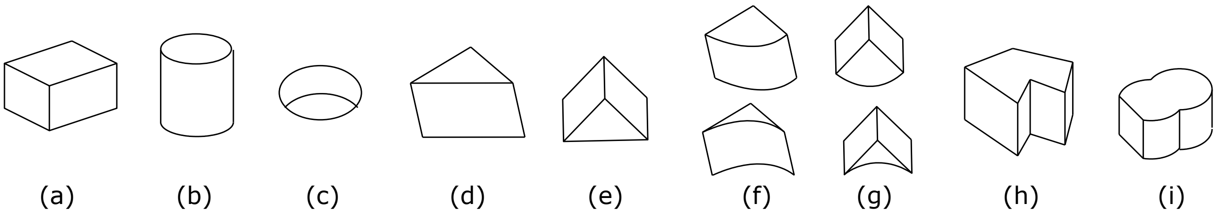


Figure 3: Nine sketch features: (a) Cuboid, (b) Cylinder, (c) Hole, (d) Concave triangular prism, (e) Convex triangular prism, (f) Two types of concave fillet, (g) Two types of convex fillet, (h) Prism, and (i) Extrusion.

3 ALGORITHM

Figure 4 shows the algorithm of this method. Each step of the algorithm is explained with Example 1 illustrated in Figure 5(a) as follows:

Step 1) When Example 1 is input to this method, intersected straight lines are first divided to line segments at their intersections. Each line segment can be recognized and classified into a straight line, an ellipse, an elliptical arc, or a Bézier curve because Example 1 is drawn using 2D CAD.

Step 2) If two lines form an L-junction, they can be extended to the nearest line. These extended lines are called additional lines and drawn as dotted lines. If two lines are connected at a tangent

point, additional lines are not created. Similarly, if three lines form a W-junction, both sides from them can be extended and two additional lines are created. In Example 1, three L-junctions and a W-junction can be detected and eight additional lines from them can be drawn as in Figure 5(b).
Step 3) If two curved line segments can be corresponding when they move in parallel and each of them has two terminals as two tangent points, a tangent line can be drawn as an additional straight line between them unless other lines are intersected. Subsequently, each face is detected by combining the lines.
Step 4) A sketch feature is detected by combining the detected sketch faces. The detection is performed in the order of holes, cylinders, cuboids, triangular prisms, fillets, prisms, and extrusions. In Example 1, a concave fillet sketch can be detected as in Figure 5(c).
Step 5) Additional lines that do not form the sketch feature detected in Step 4 are removed.

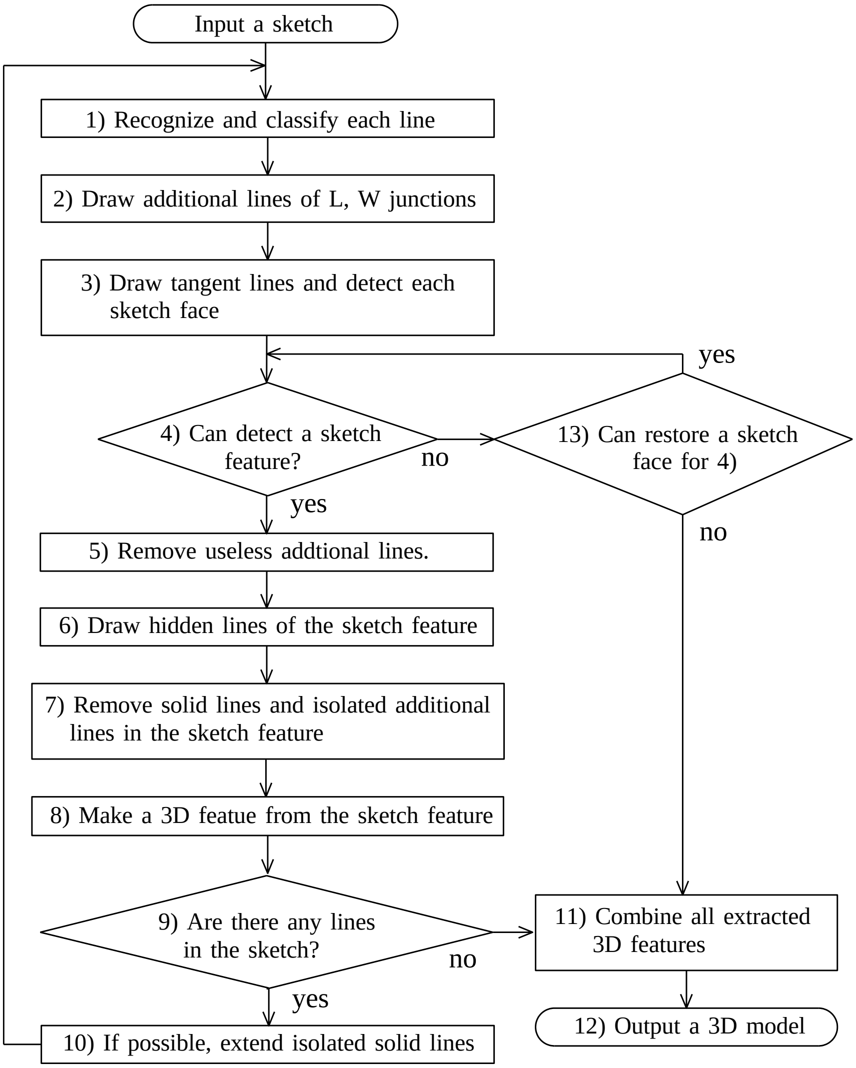


Figure 4: Algorithm of this method.

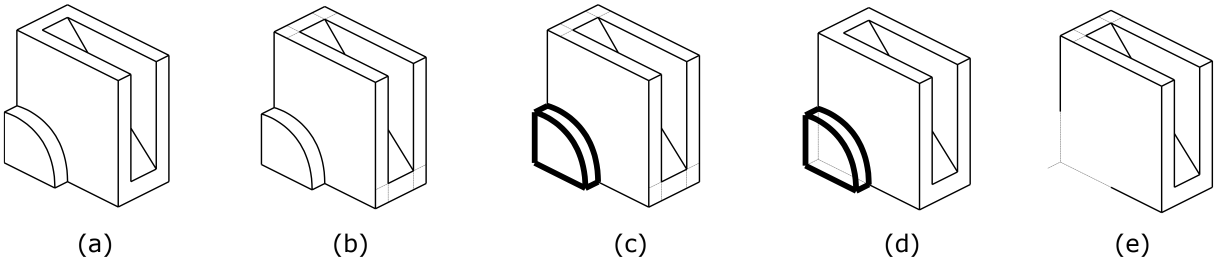


Figure 5: Example 1 and extraction of a fillet: (a) Example 1, (b) Addition of eight additional lines, (c) Detection of a concave fillet sketch, (d) Removal of useless additional lines and addition of hidden lines of the fillet, and (e) Removal of solid lines of the fillet.

Step 6) All hidden lines of the detected sketch are drawn as additional lines. In Example 1, eight additional lines are removed, and three hidden lines are drawn as in Figure 5(d).
Step 7) To extract the detected sketch feature, all solid lines of it are first removed as in Figure 5(e). Subsequently, all additional lines that are not hidden lines of the sketch feature are changed into solid lines. Furthermore, all isolated additional lines that do not form any loops of lines are removed. Finally, all additional lines that are hidden lines are changed into solid lines such as in Figure 6(a).
Step 8) The extracted sketch feature is changed into a 3D model as a 3D feature. In the figure, the 3D fillet is drawn separately.
Step 9) If there are no lines, Step 11 is executed. In Example 1, many lines remain, as shown in Figure. 5(a); therefore, Step 10 is executed.
Step 10) If an isolated solid line exists, it is extended until it is in contact with the nearest line. Subsequently, Step 1 is executed again. In Example 1, the following processes are described in the following paragraph.
Step 11) Each 3D feature is combined from the last to the first, step by step. Each combination can be performed in accordance with solid lines that are changed from additional lines between two 3D features.
Step 12) A 3D model is output as the solution of the input sketch.
Step 13) If a sketch feature cannot be detected, our automatic restoration system of sketch faces is executed. A detailed explanation is described in the following paragraph.

In Example 1, after the 3D fillet is extracted, a cuboid sketch can be detected, as in Figure 5(b). Figure 5(c) is created by performing Step 7. In this figure, four additional lines can be changed into solid lines, and two isolated solid lines can be extended and connected, as in Figure 5(d). After the 3D cuboid is extracted, a triangular prism sketch can be detected, as in Figure 6(e).

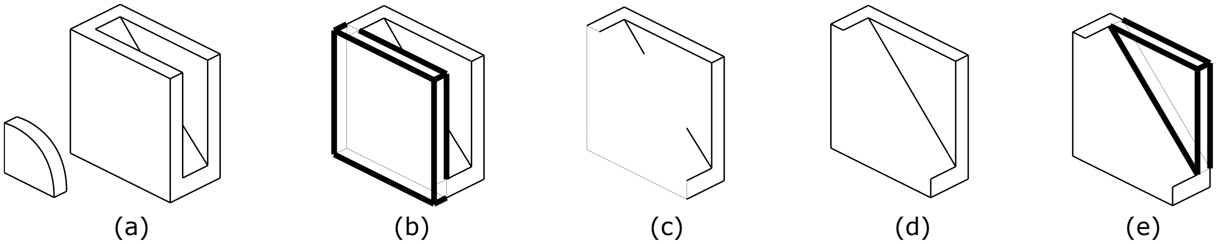
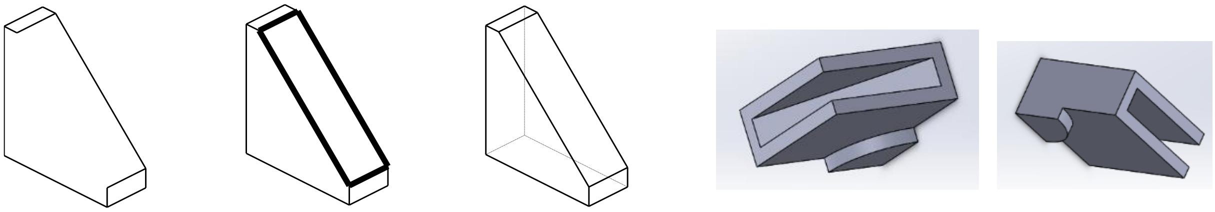


Figure 6: Extraction of a cuboid in Example 1: (a) Removal of an isolated additional line, change of two additional lines into solid lines and extracted 3D fillet, (b) Detection of a cuboid sketch, (c) Removal of solid lines of the cuboid and an isolated additional line, (d) Change and extension of lines, (e) Detection of a triangular prism sketch.

After the prism is extracted, Figure 5(a) is created. From this figure, sketch features cannot be detected. Therefore, our automatic restoration system of sketch faces is executed as follows. We have developed an inductive learning system for restoring broken sketches [14]. In this system, a parallelogram can be restored from three lines, as in Figure 5(b). In this figure, a polygon face and three parallelograms are connected to each other. Therefore, a prism sketch can be detected, as in Figure 5(c). After the prism is extracted, no lines are found in Step 9. Therefore, Step 11 is executed. In this step, the 3D prism, 3D triangular prism, 3D cuboid, and 3D fillet are combined step by step in this order. Consequently, in Step 12, the solution of Example 1 can be output, such as in Figure 5(d).



(a) (b) (c) (d)

Figure 7: Completion of 3D model in Example 1: (a) Extraction of the prism, (b) Restoration of a parallelogram, (c) Detection of a prism sketch, and (d) Two overviews of the solution.

4 EXAMPLES

In this section, three examples are provided. Figure 6(a) is Example 2 that is a sketch of a bearing housing. In Figure 6(b), all additional lines are drawn. In Figure 6(c), three-hole sketches are detected. In Figure 6(d), all solid lines of the holes are removed. In this figure, all additional lines are isolated lines. Therefore, they are removed as in Figure 6(e). Furthermore, in this figure, the three extracted holes are drawn separately. In Figure 6(f), a cylinder sketch is detected. In Figure 6(g), the cylinder is extracted and drawn separately. In Figure 6(h), a triangular prism sketch is detected. In Figure 6(i), the prism is extracted and drawn separately. Here, a curved parallelogram can be restored. Therefore, two extrusion sketches can be detected, as shown in Figure 6(j). Consequently, the solution of Example 2 can be output, as shown in Figure 6(k).

Figure 7(a) is Example 3 that is a sketch of a gear. In Figure 7(b), a hole sketch is detected. In Figure 7(c), the hole is extracted, and a prism sketch would be detected by a polygon and nine parallelograms. Generally, the detection of prisms and extrusions is difficult. Therefore, when one of them may be detected, its temporal 3D feature is created using this method. If no contradictions exist between the sketch of the 3D feature and the original sketch, the 3D feature can be extracted. Consequently, the solution of Example 2 can be output as in Figure 7(d).

Figure 8(a) is Example 4 that is a sketch of a spanner. In Figure 8(b), a hole sketch is detected. In Figure 8(c), the hole is extracted, and an extrusion sketch can be detected by a complex face, a parallelogram and two curved parallelograms. After the extrusion is extracted, only two straight lines remain, as shown in Figure 8(d). These two lines can be a mistake of the designer. Although a parallelogram can be restored from them, sketch features cannot be detected from it. Therefore, Step 11 is executed, and a 3D model is output, as shown in Figure 8(e). If the designer wishes to add those two lines as two thin slots or bosses to the 3D model, it can be performed in the original sketch. Nevertheless, this method would be useful for the designer.

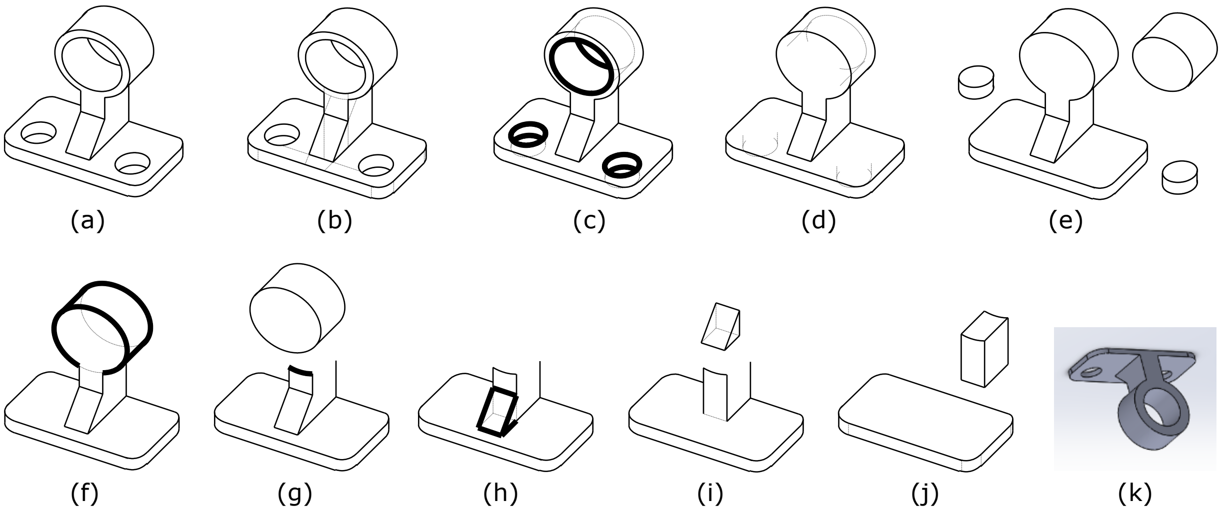


Figure 8: Example 2: (a) Example 2, (b) Addition of 15 additional lines, (c) Detection of three hole sketches, (d) Removal of solid lines of them, (e) Removal of isolated additional lines and extracted three holes, (f) Detection of a cylinder sketch, (g) Extraction of the cylinder, (h) Detection of a triangular prism sketch, (i) Extraction of the prism, (j) Detection and extraction of two extrusion sketches, and (k) Overview of the solution.

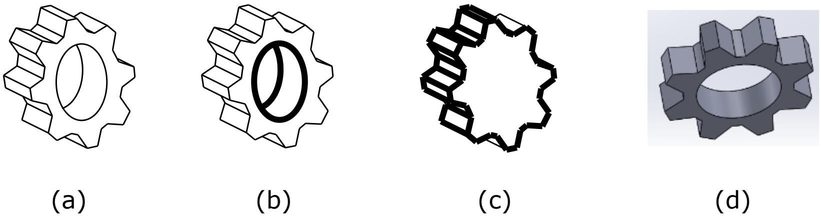


Figure 9: Example 3: (a) Example 3, (b) Detection of a hole sketch, (c) Removal of the hole and detection of a prism sketch, and (d) Overview of the solution.

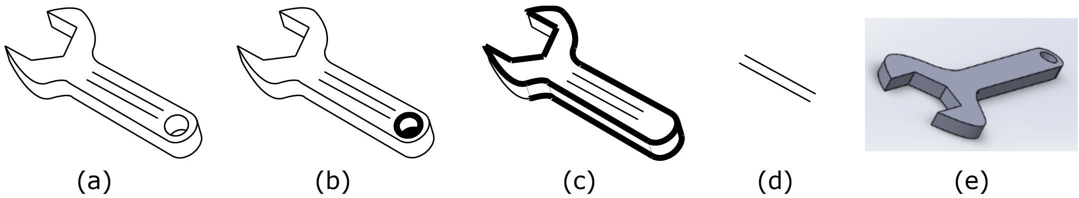


Figure 10: Example 4: (a) Example 4, (b) Detection of a hole sketch, (c) Removal of the hole and detection of an extrusion sketch, (d) Two remained lines, and (e) Overview of output 3D model.

5 DISCUSSION

This method is strongly oriented toward developing a more useful system to automatically convert sketches into 3D models. Therefore, many issues are considered, as follows. In the definition of sketch faces, numerous faces exist. For example, many types of curved complex faces, and faces with holes are required for detecting concave shapes in the sketches of 3D objects such as pallets and vessels. Furthermore, to apply pipes and tubes that often appear in mechanical objects, torus and sphere faces must be defined as sketch faces. They are also effective for detecting many types of round fillets. These extended sketch faces would enable more sketch features to be defined, and more types of sketches of 3D objects to be applied. Furthermore, Plumed, et al. [14] considered many types of 3D features for the conversion. This method can be applied only to the sketch of a mechanical object because the handling of sketches in stationery, dinnerware, animals, pictures, etc. would be more adaptive to artificial intelligence (AI) systems and/or image processing systems compared with this method. When a human views a sketch of them, he/she can apply pattern matching processes easily. Furthermore, if plural objects are sketched, this system cannot be applied. Meanwhile, the automatic conversion of rough sketches into correct ones geometrically is important. Dematapitiya et al. [7] addressed this issue and developed their own method. However, it is difficult to solve the issue perfectly because each designer draws different sketches. Therefore, the issue might become a commercial issue for us. Moreover, although an input sketch is viewed from a general view in this method, it is difficult to obtain when the shape of an object becomes complex for users. In this case, 3D models would be created as solutions in this method where each of them comprises 3D sketch features. Therefore, if a user wishes to modify a 3D model, an interactive system for the modification will be an issue in this method.

6 CONCLUSION

We proposed a method to automatically convert precise sketches drawn in 2D CAD systems into 3D models of mechanical objects. For the conversion, sketch faces were first defined. Subsequently, sketch features comprising sketch faces were defined. In this method, when a sketch was input, each sketch feature was detected and extracted step by step, and the solution could be obtained by combining all extracted 3D features. If detection is difficult, our automatic restoration system can be applied. Herein, a detailed algorithm of this method was provided, and its effectiveness and practicability were demonstrated in four examples. Finally, although many issues may arise in this method, they are discussed in detail here.

Masaji Tanaka, http://orcid.org/0000-0002-5266-9182
Motoki Terano, http://orcid.org/0000-0001-9621-4174
Tetsuya Asano, http://orcid.org/0000-0002-4988-6928

REFERENCES

[1] Cao, L.; Liu, J.; Tang, X.: What the back of the object looks like: 3D reconstruction from line drawings without hidden lines, IEEE Transaction on Pattern Analysis and Machine Intelligence, 30(3), 2007, 507–517.

[2] Cheona, S.; Kim, B. C.; Munc, D.; Hand, S.: A procedural method to exchange editable 3D data from a free-hand 2D sketch modeling system into 3D mechanical CAD systems, Computer-Aided Design, 44(2), 2012, 123-131. https://doi.org/10.1016/j.cad.2011.10.003

[3] Clowes, M.B.: On seeing things, Artificial Intelligence, 2(1), 1971, 79–116. https://doi.org/10.1016/0004-3702(71)90005-1

[4] Company, P.; Piquer, A.; Contero, M.; Naya, F.: A survey on geometrical reconstruction as a core technology to sketch-based modeling, Computers & Graphics, 29(6), 2005, 892–904. http://dx.doi.org/10.1016/j.cag.2005.09.007

[5] Company, P.; Ashley, P.; Varley, C.: A method for reconstructing sketched polyhedral shapes with rounds and fillets, Smart Graphics, Lecture Notes in Computer Science 6133, 2010, 152-155.

[6] Company, P.; Plumed, R.; Varley, C.: A fast approach for perceptually-based fitting strokes into elliptical arcs, The Visual Computer, 31(6), 2015, 775-785. http://dx.doi.org/10.1007/s00371-015-1099-6

[7] Dematapitiya, S.; Kawazoe, M.; Nishikawa, A; Sakurai, M.; Saga, S.: Snapping of fuzzy objects using the multi-resolution fuzzy grid snapping technique, Information and Media Technologies, 4(2), 2009, 463-474. https://doi.org/10.2197/ipsjjip.17.47

[8] http://help.solidworks.com/2016/English/SolidWorks/sldworks/t_Beginning_a_3D_Sketch.htm

[9] http://www.sketchup.com/

[10] Huffman, D. A.: Impossible objects as nonsense sentences, Machine Intelligence 6, 1971, 295–323.

[11] Lee, H.; Han, S.: Reconstruction of 3D interacting solids of revolution from 2D orthographic views, 37(13), 2005, 1388-1398. https://doi.org/10.1016/j.cad.2005.01.007

[12] Malik, J.: Interpreting line drawings of curved objects, International Journal of Computer Vision, 1, 1987, 73–103. http://dx.doi.org/10.1007/BF00128527

[13] Mitani, J.; Suzuki, H.; Kimura, F.: 3D sketch: Sketch-based model reconstruction and rendering, From Geometric Modeling to Shape Modeling, Springer, 2000, 85-98.

[14] Plumed, R.; Varley, P.; Company, P.: Features and design intent in engineering sketches, Studies in Computational Intelligence, 441, 2012, 77-106.

[15] Tanaka, M.; Kaneeda, T.: Feature extraction from sketches of objects, Computer-Aided Design & Applications, 12(3), 2014, 300-309. http://dx.doi.org/10.1080/16864360.2014.981459

[16] Tanaka, M.; Takamiya, Y.; Tsubota, N.; Asanuma, S.; Iwama, K.: Reconstruction of solid models from sketches including curves with inductive learning technique, Computer-Aided Design & Applications, 14(5), 2016, 632-641. https://doi.org/10.1080/16864360.2016.1273580

[17] Varley, C.; Martin, R.; Suzuki, H.: Making the most of using depth reasoning to label line drawings of engineering objects, Proceedings of The Ninth ACM Symposium on Solid Modeling and Applications, 2004, 191-202.