Isometric Conversion of Mechanical Sketches into 3D Models

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

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

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

Abstract. Sketches in the form of line drawings are important illustrations to directly express the overviews of objects, especially mechanical objects. Numerous methods that automatically convert sketches into 3D models have been proposed. However, no practical system for conversion has been developed till now. Consequently, we developed a method for this conversion. In our SFBCM (Sketch Feature-Based Conversion Method), when a sketch is input, sketch features that indicate simple sketches of objects such as cubes, cylinders are detected and extracted as 3D features step by step. Consequently, a 3D model can be obtained by combining them based on the sketch. However, the implementation of SFBCM has remained an important issue. In this paper, an algorithm for the implementation of SFBCM is proposed. For a simpler implementation, this algorithm applies human perception. For example, people prefer to draw isometric and symmetric sketches of objects. Consequently, various types of sketches can be handled in this algorithm. The effectiveness of revised SFBCM with this algorithm is demonstrated with three complex examples, and the limitations and issues are discussed in detail.

Keywords: Isometric, Sketch, Sketch Feature, 3D Model, Human Perception, SFBCM.
DOI: https://doi.org/10.14733/cadaps.2021.772-785

1 INTRODUCTION

Sketches in the form of line drawings are commonly observed in magazines, books, manuals, etc. Sketches are also important for designers, especially mechanical designers, when they invent new ideas of products and their parts. The automatic conversion of sketches into 3D models will be advantageous for several applications. Moreover, it is expected that robots will be able to understand sketches by their converted 3D models in the future. In the last fifty years, numerous methods to automatically convert sketches into 3D models have been considered and developed. However, no real system for the conversion has been developed till now. We have been developing a method for the conversion of sketches to 3D models for approximately seven years. Consequently, we proposed a fundamental algorithm to achieve this conversion [23-24]. However, the implementation of this algorithm has been ignored owing to is significant complexity. Therefore, this is an important challenge in our method.

Recently, when we retrieved various types of sketches from several websites, we identified an important cue for the implementation. Let us assume that there is a pole. When a person draws its sketch on a paper, he/she will draw a sketch as shown in Fig. 1(a) rather than a sketch as shown in Fig. 1(b), because he/she wants to emphasize that the object is long. Therefore, people generally prefer drawing isometric sketches of objects. Fig. 2 shows three sketches of a cube. Fig. 2(a) shows a precise sketch, which is a copy of a screen in a solid modeler (SolidWorks). Fig. 2(b) shows an isometric sketch. Fig. 2(c) shows a sketch that is not precise but understandable for people. These three sketches will be recognized as the same cube by people because cubes are more familiar structures than rhombus-shaped objects. In addition, it can be assumed that people generally consider most isometric objects as 2D pictures, except when they are placed before their eyes. For example, Fig. 3 shows pictures of a cardboard box. In Fig. 3(a), we captured a picture using a camera. In Fig. 3(b), the edges of Fig. 3(a) are colored. Although the correct lengths of the red, blue, and green lines are 275, 205, and 80 mm, respectively, certain limitations in this figure are obvious; for example, the lengths of the three red lines are different, and two blue lines are not parallel. Generally, people seldom look at a box, as shown in this figure. In Fig. 3(c), we captured a picture when the camera is placed at a distance of 1000 mm from the box. In this figure, it can be observed that the box appears to be similar to a cuboid. Moreover, it is a normal canonical view (e.g. [10],[18]). In addition, it can be assumed that people can watch televisions and movies as 3D worlds on 2D screens. We observed that such human perceptions can aid in achieving an easy implementation of our method.

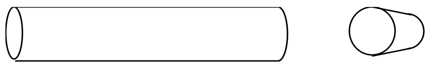


Figure 1: Two sketches of a pole: (a) Isometric sketch and (b) Perspective sketch.

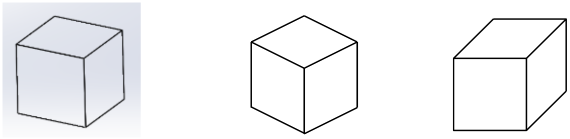


Figure 2: Three sketches of a cube: (a) Precise sketch, (b) Isometric sketch, and (c) Another sketch.

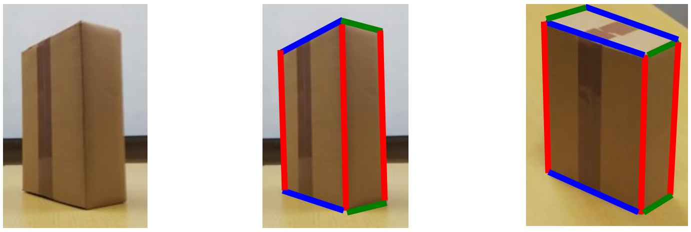


Figure 3: Three pictures of a cardboard box: (a) Box is positioned in front of the camera, (b) Comparison of the edges for the image illustrated in (a), and (c) Normal view.

In this paper, an algorithm for implementing the proposed method is proposed. This algorithm is based on the aforementioned concept of human perception. In Section 2, the related works are presented. In Section 3, the principle of this algorithm is explained. In Section 4, this algorithm is explained in detail with three examples. In Section 5, the effectiveness and limitations of this algorithm are discussed.

In this algorithm, sketches are drawn using 2D CAD systems and/or graphics processors on tablets, laptop PCs, etc. The process required to handle sketches drawn on paper is separated from this algorithm because it is an issue in image processing. Although rough sketches may be handled in future works, only accurate sketches are handled in this paper. For example, a parallelogram must form two pairs of lines whose lengths are the same and are parallel, and there are four vertices. It is to be noted that, in this paper, a 2D closed contour that contains no lines is called a region and a 3D contour is called a face. A sketch feature converted into a 3D model is called a 3D feature.

2 RELATED WORKS

There are numerous papers on the automatic recognition of sketches in the form of line drawings and/or their conversion into 3D models. Their classification is presented in [5]. The original line labeling technique was developed as Huffman-Clowes labeling, [4],[13]. In this labeling process, the objects of the sketches were limited to opaque trihedral polyhedrons, and a sketch was an orthogonal projection of an object viewed from a general position. Each line segment of a sketch was labeled as "+" (convex line), "-" (concave line), or with an arrow (occluding line). From the labeling, the vertices were classified into the following four types of junctions: *L*, *W*, *T*, and *Y*-junctions. This naming was derived from the shapes of the alphabets, i.e., "L," "W," "T," and "Y," respectively. The relationships between the labeling and the junctions were summarized as a junction dictionary. Fig. 4 shows a sample of line labeling. Fig. 4(a) shows a sample sketch of polyhedrons. In Fig. 4(b), each line segment is labeled. Arrowed, "+," and "-" lines are colored blue, red, and green, respectively. From this figure, each junction can be recognized, as shown in Fig. 4(c), by using the junction dictionary. The three red points are *Y*-junctions, and two of them express convex corners because each of them consists of only "+" lines, and the other is a concave corner. Five green points, six blue points, and a brown point express *W-*, *L-*, and *T*-junctions, respectively.

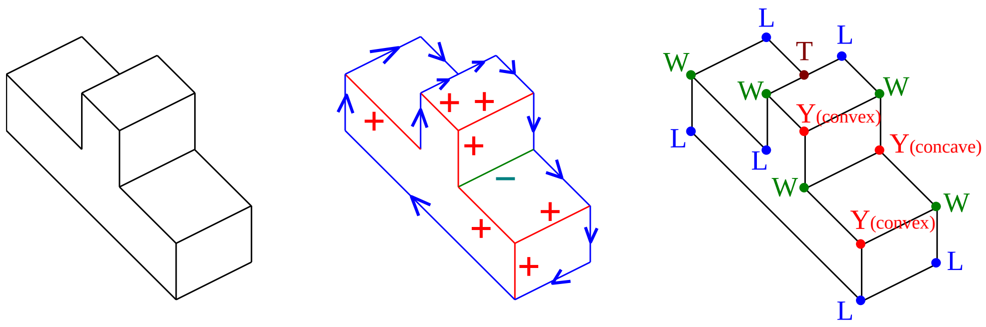


Figure 4: Sample of line labeling: (a) Sample sketch, (b) Line labeling, and (c) Junctions.

Kirousis [16] indicated the computational complexity of line labeling and presented the following proposition, "If a sketch is realizable, then it is labelable." Moreover, Varley et al. [27] handled high-order junctions such as *K*- and *X*-junctions. Although these labeling techniques were applied to sketches consisting of only straight lines, Malik [17] created a junction dictionary for curved lines in sketches. For a practical conversion, classically, skewed symmetry theory proposed by Kanade [15] and linear equations-based technique proposed by Sugihara [21] are important. These techniques are applied for practical conversion, as demonstrated in [8]. Moreover, Varley et al. [27] identified a "cubic corner" based on [19], which is effective for practical conversions. Ribs, pockets, and fillets in sketches were researched in [6],[20].

In recent years, Company et al. [7] investigated techniques to detect junctions from sketches using human perception. From this analysis, it can be determined that human perception is important for conversion, which is incorporated in this algorithm. Interactive or semiautomatic systems for conversion were previously proposed, [1],[30]; however, there are no fully automatic systems for conversion. Further, 3D sketching systems were also developed [11-12]. Although these systems might be useful for beginners, they are only a few types of solid modelers, for example, CATIA and SolidWorks. Fundamentally, sketches in the form of line drawings are perceived as freehand drawings of objects for people rather than for systems. Neural network techniques have also been actively used for conversion [2-3],[9],[14],[26],[28-29]. However, a learning process is required for conversion in these techniques. The learning is suitable for known objects such as tables, chairs, cups; however, this technique is not suitable for different types of mechanical and creative objects because each of them must be learned individually and repeatedly for different cases. Therefore, the learning of these objects is wasteful. In summary, although all the studies described above are effective for conversion, the types of convertible objects are strictly limited. Our proposed algorithm in this paper aims to extend the limitation of convertible objects significantly more than conventional techniques.

3 PRINCIPLE OF THE PROPOSED ALGORITHM

3.1 Sketch Features and Example 1

In our method, all primitive sketches are initially defined as "sketch features." Here, sketch features are different from "machining features" because our idea of sketch features is based on the automatic conversion of orthographic views into 3D models, as presented in [22]. Therefore, we have not referred to machining features and their recognition techniques. This is because determining several more sketch features for conversion is a more important problem than recognizing them in the present step of our method. Fig. 5 shows three basic sketch features that indicate a cuboid, cylinder, and round hole. Most people can recognize and draw these sketches because they are familiar and commonly drawn figures. In our method, a sketch is converted into a 3D model by extracting sketch features sequentially from the sketch. Further, a sketch is an orthogonal and opaque projection of a 3D object placed in a general position. Each sketch consists of ellipses, elliptical arcs, and straight lines.

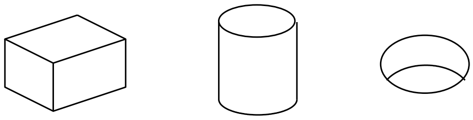


Figure 5: Basic three sketch features: (a) Cuboid, (b) Cylinder, and (c) Round hole.

Here, our method is explained simply using Example 1, as shown in Fig. 6(a). When Example 1 is input, first, straight lines are divided at their intersections. Second, additional lines are drawn as dotted lines. They can be drawn from *T*-, *W*-, and *L*-junctions, such as extending their lines within the input sketch. If a curved line is an element of a *T*-junction, it is extended in a manner similar to forming an oval. If an additional line cannot become a part of a closed loop of lines, it is cut until the loop is formed. In Fig. 6(b), an additional line (green) is drawn from a *T*-junction. Third, a basic sketch feature can be detected, such as a cylinder in Fig. 6(c). In this figure, the detected cylinder is blue and the hidden arc is in green. Fourth, the cylinder is extracted as a 3D cylinder from the input sketch, as shown in Fig. 6(d). In this algorithm, hidden line(s) become additional line(s) after a 3D feature is extracted. After the extraction, an additional arc in the remaining sketch can be extended to make an (pink) ellipse. The ellipse can become the contact face to the bottom (pink) face of the 3D cylinder, as shown in this figure. Subsequently, a sketch feature of a cuboid can be detected as a 3D cuboid, as shown in Fig. 6(e). Then, if any sketch features cannot be detected, a restoration process is executed to create a sketch feature using this algorithm. A solid model of Example 1 can be obtained by combining the 3D cuboid and 3D cylinder at their contact face. A detailed explanation of this process is provided in the next subsection. Fig. 7 shows a summarized flowchart of the method described above. Here, we call our method SFBCM (Sketch Feature-Based Conversion Method). In this figure, Step 6 implies that each additional line forming a *T*-junction is removed because the line must be occluded from the other line(s) so that it cannot form any sketch features. A detailed explanation of the flowchart can be obtained in [23-24].

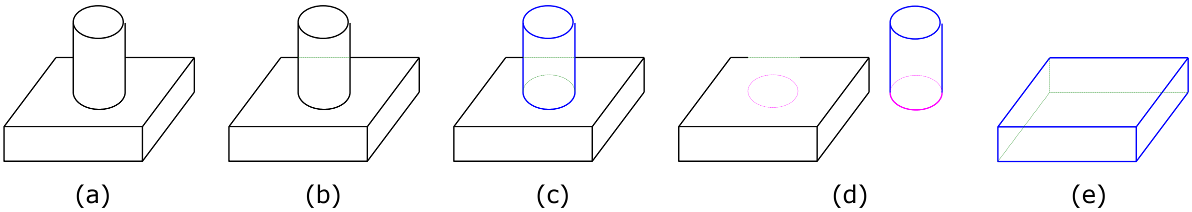


Figure 6: Example 1: (a) Example 1, (b) Additional line is drawn, (c) Cylinder is detected, (d) Cylinder is extracted as a 3D cylinder and a contact face is drawn, and (e) Cuboid is detected.

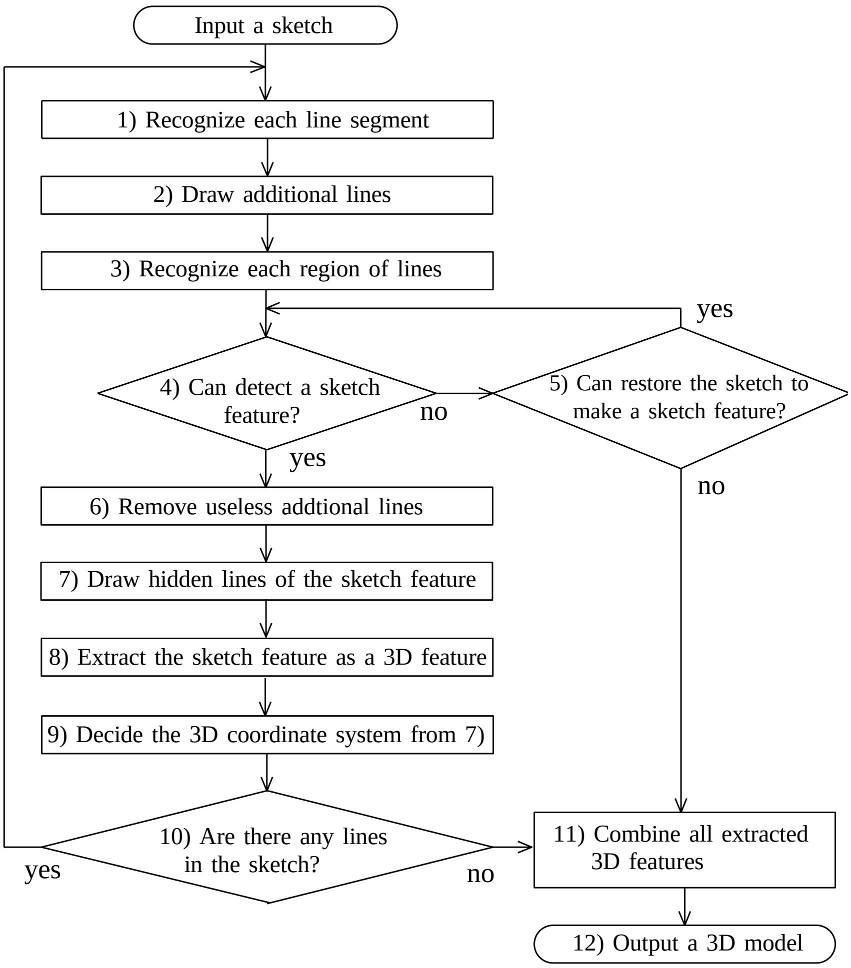


Figure 7: Flowchart of our method.

3.2 Relationship between Sketch Features and Their 3D Models

When a sketch of a cuboid is provided, it is difficult to determine the depth or height of each vertex to make a 3D model. A "cubic corner" (refer [27]) is effective for overcoming this challenge. Fig. 8 illustrates a cubic corner. A *Y*-junction of three lines (*VA*, *VB*, and *VC*) is drawn in the *x*-*y* coordinate system. *E*, *F*, and *G* indicate angles *BVC*, *AVC*, and *AVB*, respectively. If the *Y*-junction forms a cubic corner in 3D space, the *z*-coordinate value of *A* (Z_A) can be calculated using the following equation.

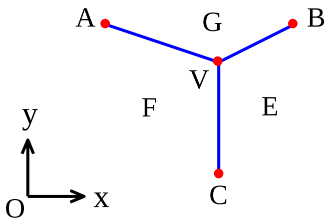


Figure 8: Explanation of "cubic corner."

$$|Z_A - Z_V| = m_{VA}/\sqrt{(\tan F \tan G) - 1} \quad (1)$$

Here, the value of Z_V can be zero. m_{VA} is the length of *VA*. Similarly, Z_B, Z_C, m_{VB}, and m_{VC} can be calculated. Thus, a 3D cuboid model can be obtained. Although the equation is precise and useful for the conversion of sketches into 3D models, there are certain problems. In Example 1, if the equation is applied, *tan*(*π*/2) is obtained; consequently no values can be calculated. However, people can observe the cuboid sketch in Example 1 because of the concept of human perception described in Section 1. Moreover, people prefer symmetrical drawings to asymmetrical ones. For example, Fig. 9 shows three sketches of a cylinder on a plate, similar to Example 1. They are copies of a screen in a solid modeler (SolidWorks). In Fig. 9(a), a cylinder stands vertically on a plate. In Fig. 9(b), a cylinder appears to lean to one side; however, the object appears similar to the sketch in Fig. 9(a) when the 3D model of Fig. 9(b) is rotated in the solid modeler, as shown in Fig. 9(c). In general, people view Figs. 9(a) and 9(c) identically, and they seldom consider that Fig. 9(c) is another view of Fig. 9(b). In addition, the manufacturing processes of symmetrical 3D objects are familiar, especially machining and assembly, and the sketch features of such mechanical objects are commonly observed. Consequently, we handle sketches of only mechanical objects for the conversion and review if each sketch is isometric and basically symmetric.

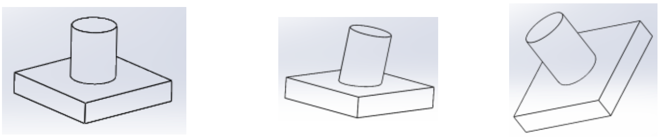


Figure 9: Three sketches of a cylinder on a plate: (a) Vertical cylinder on a plate, (b) Cylinder leaning to one side, and (c) Another view of (b).

Therefore, in this algorithm, the definition of a cuboid sketch considers that three parallelograms share three edges that form a *Y*-junction. Therefore, the "cubic corner" is not applied, and the lengths of three lines forming the *Y*-junction are regarded as the correct lengths of three edges in the 3D model of the cuboid sketch. In our experimental system, when a sketch of a cuboid is input, as shown in Fig. 10(a), a 3D cuboid model is obtained as shown in Fig. 10(b). As a result, our SFBCM becomes simpler in the aspect of implementation than conventional methods such as "cubic corner." Moreover, for simpler positioning of the 3D cuboid, a 3D coordinate system is placed as shown in Fig. 11(a) based on Fig. 8 by using three edges to form a cubic corner. In this figure, Point *C* in Fig. 8 corresponds to the origin. Further, *VA*, *VB*, and *VC* are parallel to the *y*-, *x*-, and *z*-axes, respectively. Therefore, the orthographic view of the cuboid can be drawn as shown in Fig. 11(b). This coordinate system enables a clearer handling of 3D objects consisting of multiple 3D features. In Example 1, a cylinder stands on a plate vertically according to human perception. Therefore, the axis of the 3D cylinder model can be regarded to be parallel to the *z*-axis. In addition, the height of the cylinder can be regarded to be identical to its sketch, and its diameter can be regarded to be identical to the longer axis of the ellipse in the sketch. Moreover, the positioning of the 3D cylinder on the 3D plate can be regarded as the center of the plate from Fig. 6(a). Consequently, the 3D form of Example 1 can be obtained as shown in Fig. 12(a) and the orthographic view of can be drawn as shown in Fig. 12(b).

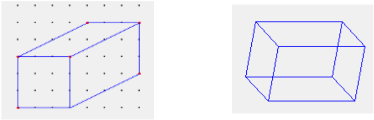


Figure 10: Automatic conversion in our experimental system: (a) Input cuboid sketch and (b) Converted 3D cuboid from (a).

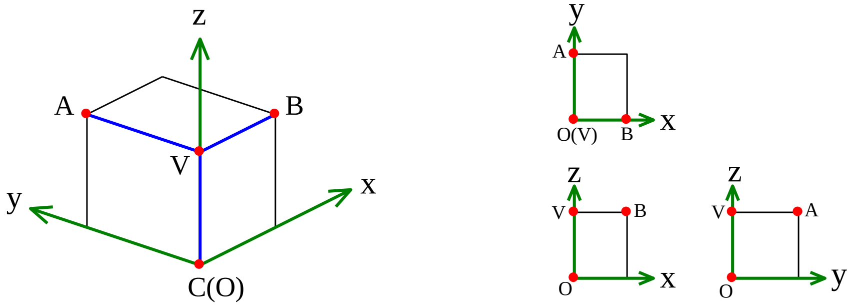


Figure 11: 3D coordinate system of a cuboid sketch formed based on the flowchart in Fig. 7: (a) Overview and (b) Orthographic view.

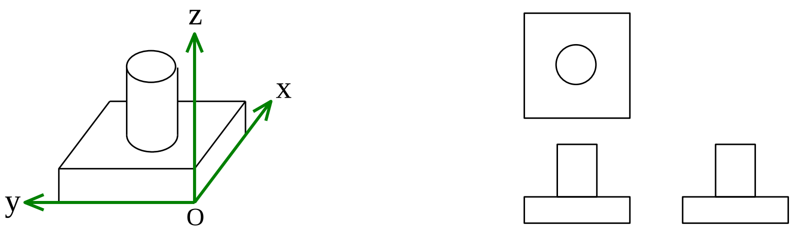
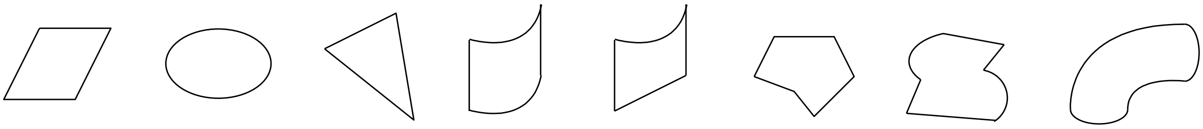


Figure 12: 3D coordinate system of Example 1: (a) Overview and (b) Orthographic view.

4 PROPOSED ALGORITHM

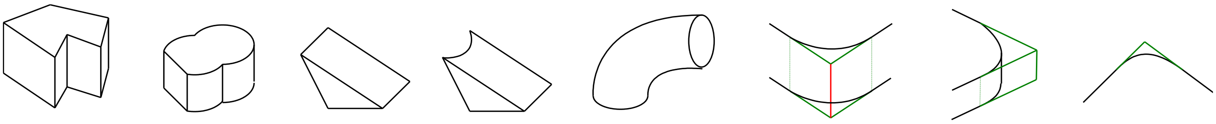
4.1 Extension of Sketch Features

Although three sketch features are illustrated in Fig. 5, to handle sketches that are more complex than Example 1, more sketch features are necessary. In our present step, eight sketch features are added; moreover, eight regions are defined for these features. Fig. 13 shows the regions, which consist of the following: parallelogram, ellipse, triangle, curved parallelogram, concave parallelogram, polygon, multi-region shape, and piped region. For example, the cylinder sketch shown in Fig. 5(b) can be defined as an ellipse with a curved parallelogram sharing half of the ellipse. Fig. 14 shows the eight sketch features. Fig. 14(a) shows a polygonal extrusion. It consists of a polygon and multiple parallelograms. Each line segment of the polygon becomes a line segment of a parallelogram, and two adjacent parallelograms share a line segment. Fig. 14(b) shows a shape with multiple extrusions. It consists of a multi-region shape and parallelogram(s) and curved parallelogram(s) similar to the polygonal extrusion. Fig. 14(c) shows a rib that consists of a triangle and parallelogram. Fig. 14(d) shows a round rib that consists of a triangle and concave parallelogram. Fig. 14(e) shows a type of pipe. There are several types of pipe sketches. Although this figure consists of a pipe face and an ellipse, the ellipse can be a half-elliptical arc. In mechanical objects, fillets are often applied to sharp corners. Thus, three types of fillets are defined as follows. Fig. 14(f) shows a front fillet. It consists of four straight lines and two curved lines, each of which contacts tangentially, as shown in this figure. From the four tangent points of these lines, two green dotted lines can be drawn as additional lines, each of which is effective for detecting the regions shown in Fig. 13. Moreover, the four green extension lines from the four straight lines can be drawn as shown in this figure. In addition, a red line can be drawn between two intersections of the four green lines. Consequently, a front fillet sketch can be changed into a polygonal corner. Fig. 14(g) shows a side fillet. Fig. 14(h) shows a hidden fillet. Both these fillets can be changed into polygonal corners using the same method used with the front fillet. We determined these sketch features from the operations in solid modelers.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 13: Eight regions: (a) Parallelogram, (b) Ellipse, (c) Rectangle, (d) Curved parallelogram, (e) Concave parallelogram, (f) Polygon, (g) Multi-region shape, and (h) Piped region.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 14: Eight sketch features: (a) Polygonal extrusion, (b) Shape with multiple extrusions, (c) Rib, (d) Round rib, (e) Pipe, (f) Front fillet, (g) Side fillet, and (h) Hidden fillet.

4.2 Three Examples

From the definitions of sketch features shown in Fig. 14, our proposed algorithm of SFBCM can be applied to more complex sketches. In this subsection, three complex examples are presented. Fig.

15(a) shows Example 2, which is a wire bracket. When the sketch is input to this algorithm, first, straight lines are cut at their intersections. In Fig. 15(b), the 10 red points indicate *L*-, *W*-, or *T*-junctions and are used to draw additional lines. The 18 blue points in this figure are tangent points between straight lines and curved lines. Some of them can be used to draw additional lines as border lines between the parallelograms and curved parallelograms. In Fig. 15(c), 13 additional lines are drawn as dotted green lines from those points. From this figure, two round holes (blue) can be detected, as shown in Fig. 15(d). In Fig. 15(e), they are extracted as f_1 and f_2. In this figure, pink ellipses are considered as the contact faces, similar to the process performed with Example 1. Further, based on human perception, it is determined that the axes of the holes must be parallel to the *z*-axis, such as the arrow of f_2. Then, four fillets are detected as f_3, f_4, f_5, and f_6 in Fig. 15(f). In Fig. 15(g), these fillets are extracted as polygonal corners and then a cuboid sketch is detected. Further, in this figure, the *x*-*y* coordinate system can be defined from the cuboid. Therefore, in this algorithm, the 3D coordinate system of a sketch can be defined from a cuboid sketch. If there are no cuboid sketches in an input sketch, it would be difficult to perform this operation. In Fig. 15(h), the cuboid is extracted as f_7. The contact face between f_7 and the other 3D feature is indicated as a green parallelogram in this figure. After the extraction, a shape with multiple extrusions can be detected, as shown in Fig. 15(i). Although this feature consists of a parallelogram, a curved parallelogram, and three regions, when the feature is converted into a 3D feature, it can be confirmed that the existence of the three regions is correct. Consequently, after the combining process is executed from f_8 to f_1, a 3D model can be obtained as shown in Fig. 15(j).

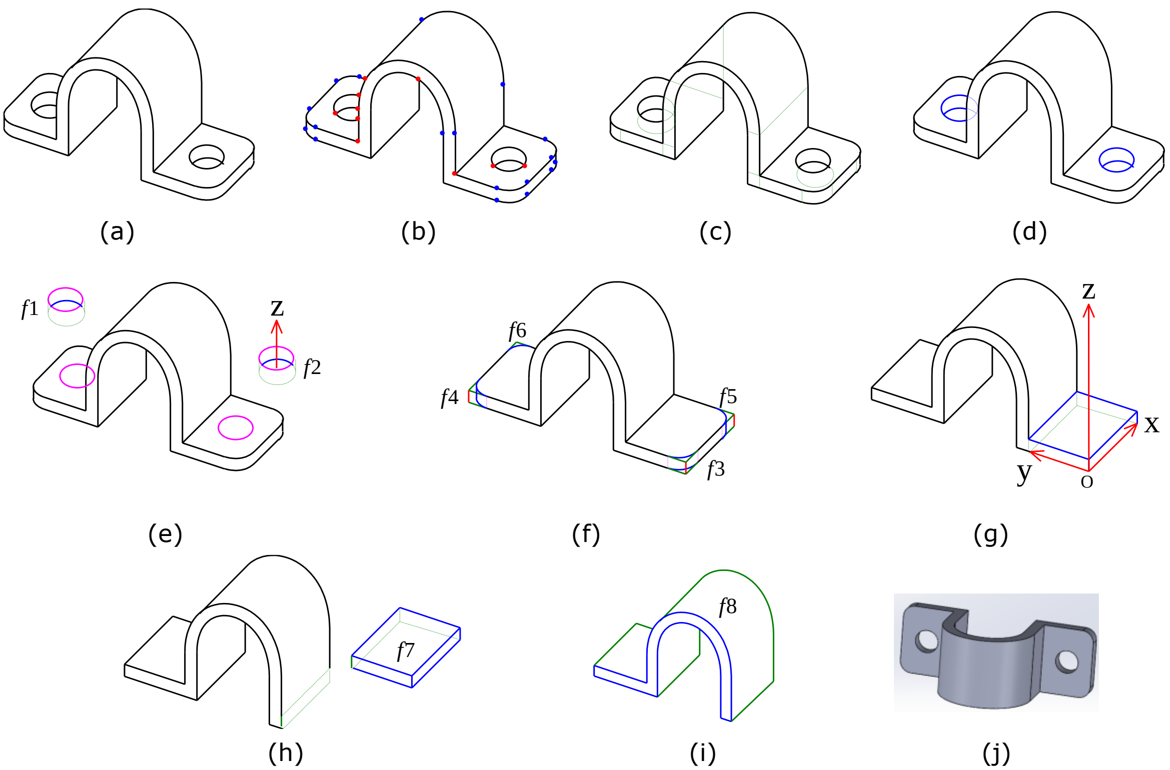


Figure 15: Example 2: (a) Example 2, (b) Red and blue points that can form additional lines, (c) Additional lines, (d) Detection of two round holes, (e) Extraction and coordination of the holes, (f) Detection of four fillets, (g) Detection and coordination of a cuboid, (h) Extraction of the cuboid as f_7, (i) Detection of a shape with multiple extrusions, and (j) Overview of the solution.

Fig. 16(a) shows Example 3, which is an eye plate. In Fig. 16(b), 11 additional lines are drawn as green dotted lines, and a pipe sketch can be detected. In Fig. 16(c), the pipe is extracted as f_1, and two contact faces as pink ellipses are indicated. Here, the *z*-axis can be defined from its cylindrical parts. In Fig. 16(d), four round holes are detected. It is determined that each hole is in the form of a tapered hole because the sizes of the ellipses on both sides are different. The recognition of the taper sketch can be inferred easily from the defined sketch feature of cylinders, and it is determined that a sketch feature of the taper has to be defined in this algorithm. Similarity, new sketch features can be identified and defined from the handling of new complex sketches. In Fig. 16(e), the tapered holes (f_2, f_3, f_4, f_5) are extracted and the *z*-axis is defined in f_2. Four pink ellipses indicate their contact faces. In Fig. 16(f), three fillets can be detected as f_6, f_7, and f_8. In Fig. 16(g), they are extracted, and a cuboid sketch is detected as f_9. The 3D coordinate system of Example 3 can be defined completely from the cuboid. Consequently, after the combining process is executed from f_9 to f_1, a 3D model can be obtained, as shown in Fig. 16(h). In this model, although one corner of the plate is not round but polygonal, it would not be difficult for a user to form a fillet in the corner. In addition, it is natural that multiple solutions can be generated from a sketch because it is a general view of a 3D object.

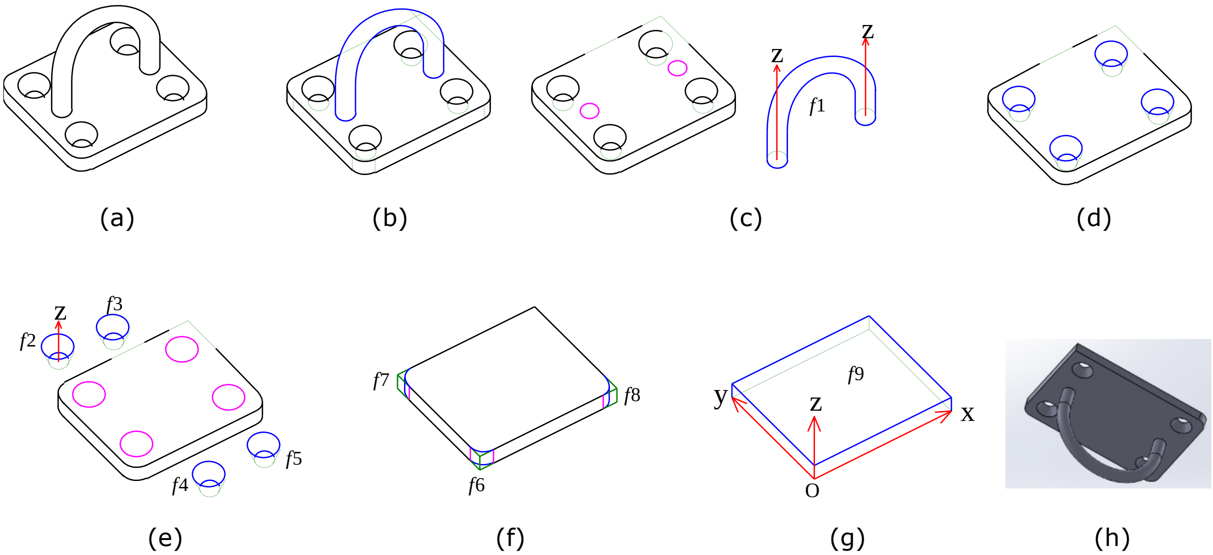


Figure 16: Example 3: (a) Example 3, (b) Additional lines and detection of a pipe, (c) Extraction and coordination of the pipe (f_1), (d) Detection of four round holes, (e) Extraction and coordination of the holes, (f) Detection of three fillets, (g) Detection and coordination of a cuboid, and (h) Overview of the solution.

Fig. 17 shows Example 4, which is an arm part of a mechanical product. This example was obtained from a textbook on AutoCAD and is shown in Fig. 17(a). However, we simplified it as shown in Fig. 17(b) because there are unnecessary hidden lines, and too many fillets that are difficult to handle for this algorithm at the present stage. First, three round holes can be detected, as shown in Fig. 17(c). In Fig. 17(d), they are extracted, and a round rib can be detected. When the rib is extracted, a shape with multiple extrusions can be detected, as shown in Fig. 17(e). In this figure, all additional lines are drawn to clarify this detection. After the extraction, a cylinder can be detected, as shown in Fig. 17(f). When it is extracted, the remaining lines can be expressed as shown in Fig. 17(g). In this figure, it is determined that all pink lines form *T*-junctions. Although it is impossible to detect any sketch features in this figure, our automatic restoring system might make a rib, as shown in Fig. 17(h). When it is extracted, another shape with multiple extrusions might be detected, as shown in Fig. 17(i). Consequently, a 3D model of Example 4 can be obtained, as shown in Fig. 18(a). However, when the model is marginally rotated using a solid modeler, it can be observed that the restored parts are separated from another solid, as shown in Fig. 18(b). Fig. 18(c) clearly shows this gap. Despite this gap, our restoring system might extend the parts to the solid. Therefore, the solution of Example 4 can be obtained as shown in Fig. 18(d). Further, the two shapes with multiple extrusions are placed at a right angle to the axis of the cylinder, and their angle to the axis is also a right angle in 3D space because the *x*-*y* coordinates cannot be defined correctly from this example.

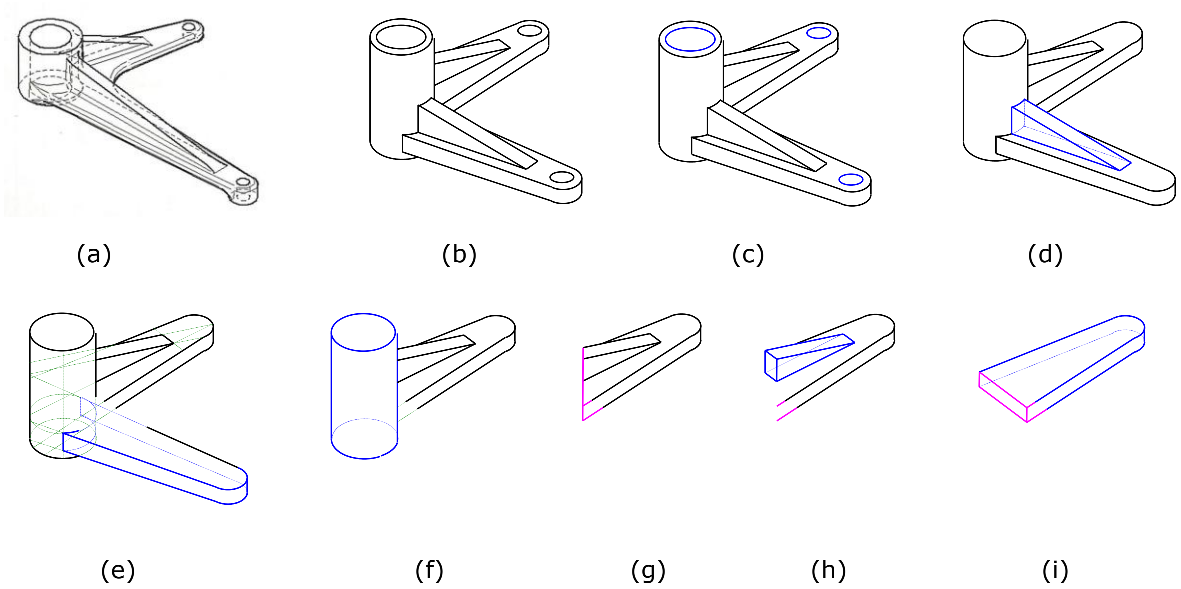


Figure 17: Example 4: (a) Original sketch, (b) Example 4, (c) Detection of three round holes, (d) Detection of a round rib, (e) Detection of a multiple extrusion, (f) Detection of a cylinder, (g) Extraction of the cylinder, (h) Restoration of a rib, and (i) Restoration of another multiple extrusion.

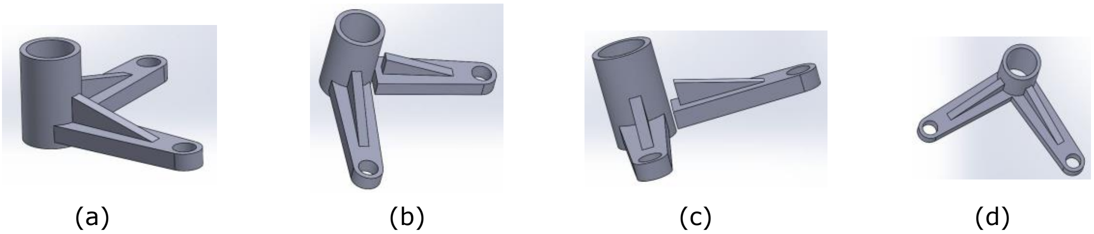


Figure 18: Final restoration in Example 4: (a) Solution created from Fig. 17, (b) Rotation of the solution, (c) Gap between two solids, and (d) Restoration of the gap.

5 DISCUSSION

This algorithm of SFBCM is based on familiar sketch features, especially for designers. We assumed that people prefer to draw isometric sketches. Therefore, symmetric sketches, such as those shown in Fig. 10(a), can be handled in this algorithm. If a cuboid sketch is detected in an input sketch, the

3D coordinate system of the sketch can be formed easily and it enables the combination of all detected 3D features to obtain the solution as a 3D model. However, there are several issues in developing an actual system for the conversion. For example, Fig. 17(a), which removes dotted lines, cannot be handled in this algorithm, but it is not difficult for skillful designers to draw this feature. The issues for our future work can be summarized as follows.

(1) From an input sketch, the processes required to decide a 3D coordinate system, detect and extract sketch feature(s) including a procedure of extraction, and assemble 3D features have to be clarified.

(2) A dictionary of sketch features must be organized. The dictionary should include the classification and more detailed definitions of sketch features for more valid detections of them from sketches. Further, the limitation of convertible sketches in our SFBCM is derived from whether an input sketch consists of only defined sketch features. Therefore, more necessary and sufficient sketch features must be obtained to expand the dictionary.

(3) The automatic restoration system for forming sketch features is the weakest point in this algorithm. Based on our past research [25], we determined that although an inductive learning system was proposed, learning techniques were useless for developing an actual system for the conversion more quickly, such as recent works related to artificial intelligence techniques, as described in Chapter 2. Therefore, learning techniques are removed from SFBCM.

(4) Rough sketches such as those shown in Fig. 17(a) can be handled for the varying requests of designers.

(5) Although we have been developing a system for conversion based on the requirements of a company handling CAD systems, a more detailed search to identify actual needs for our system would be necessary. This search will be influential in solving the above issues more efficiently.

6 CONCLUSION

In this paper, an algorithm was proposed for the implementation of our method (SFBCM) that automatically converts sketches in the form of line drawings into 3D models. The results are summarized as follows:

• Eleven sketch features (eight new and three basic features) were defined; moreover, eight regions were defined for the conversion.

• The concept of human perception was introduced to simplify the implementation. In particular, it was assumed that people prefer to draw isometric and symmetric sketches as they can be more effective. In future, we plan to determine more useful assumptions in human perception.

• It was determined that a 3D coordinate system can be fixed from a cuboid sketch. Therefore, combinations of detected sketch features could be easily converted.

• Three complex examples were shown to indicate the effectiveness of this algorithm.

• Five issues of this algorithm were discussed in detail in Section 5.

Masaji Tanaka, https://orcid.org/0000-0002-5266-9182
Tetsuya Asano, https://orcid.org/0000-0002-4988-6928

REFERENCES

[1] Bobenrieth, C.; Cordier, F.; Habibi, A.; Seo, H.: Descriptive: Interactive 3D Shape Modeling from A Single Descriptive Sketch, Computer-Aided Design, 128(102904), 2020, http://doi.org/10.1016/j.cad.2020.102904

[2] Chen, Q.; Nguyen, V.; Han, F.; Kiveris, R.; Tu, Z.: Topology-Aware Single-Image 3D Shape Reconstruction, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2020, 270-271, doi.org/10.1109/CVPRW50498.2020.00143

[3] Choy, C; Xu, D.; Gwak, J.; Chen, K.; Savarese, S.: 3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction, Computer Vision – ECCV, 2016, 628-644, https://doi.org/10.1007/978-3-319-46484-8_38

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

[5] 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

[6] 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.

[7] Company, P.; Plumed, R.; Varley, P. A. C.; Camba, J. D.: Algorithmic Perception of Vertices in Sketched Drawings of Polyhedral Shapes, ACM Transactions on Applied Perception, 16(3), 2019, Article 18, https://doi.org/10.1145/3345507

[8] Grimstead, I. J.; Martin, R. R.: Creating solid models from single 2D sketches, Proceedings of the third ACM symposium on SMA '95, 1995, 323–337, https://doi.org/10.1145/218013.218082

[9] Guo, T.; Cui, R.; Qin, X.; Wang, Y.; Tang, Z.: Bottom-up/top-down geometric object reconstruction with CNN classification for mobile education, PG '18: Proceedings of the 26th Pacific Conference on Computer Graphics and Applications, 2018, 13-16,

[10] Hall, P.; Owen, M.: Simple canonical views, Proc. BMVC, 2005, 839–848.

[11] http://help.solidworks.com/2021/English/SolidWorks/sldworks/c_3d_sketching_top.htm

[12] https://www.sketchup.com/

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

[14] Jiang, L.; Shi, S.; Qi, X.; Jia, J.: GAL: Geometric Adversarial Loss for Single-View 3D-Object Reconstruction, Proceedings of the European Conference on Computer Vision (ECCV), 2018, 802-816, http://doi.org/10.1007/978-3-030-01237-3_49

[15] Kanade, T.: Recovery of the Three-Dimensional Shape of an Object from a Single View, Artificial Intelligence, 17, 1981, http://doi.org/10.1016/0004-3702(81)90031-X

[16] Kirousis, L. M.; Papadimitriou, C. H.: The complexity of recognizing polyhedral scenes, Journal of Computer System Sciences, 37(1), 1988, 14–38, https://doi.org/10.1016/0022-0000(88)90043-8

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

[18] Niimi, R.; Yokosawa, K.: Three-quarter views are subjectively good because object orientation is uncertain, Psychonomic Bulletin & Review, 16, 2009, 289–294, http://doi.org/10.3758/PBR.16.2.289

[19] Perkins, D. N.: Cubic Corners, Quarterly Progress Report 89, MIT Research Laboratory of Electronics, 1968, 207–214.

[20] Plumed, R.; Company, P.; Varley, P. A. C.; Martin, R.: From Sketches to CAM Models: Perceiving Pockets and Steps in Single-View Wireframe Sketches of Polyhedral Shapes, Proc. 2013 ACM Conf. Pervasive and Ubiquitous Computing Adjunct Publication, 2013, 951-958, http://dx.doi.org/10.1145/2494091.2499207

[21] Sugihara, K.: Machine Interpretation of Line Drawings, MIT Press, 1986.

[22] Tanaka, M.; Iwama, K.; Hosoda, A.; Watanabe, T.: Decomposition of a 2D Assembly Drawing into 3D Part Drawings, Computer-Aided Design, 30(1), 1998, 37-46, https://doi.org/10.1016/S0010-4485(97)00051-1

[23] 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

[24] Tanaka, M.; Terano, M.; Asano, T.; Higashino, C.: Method to Automatically Convert Sketches of Mechanical Objects into 3D Models, Computer-Aided Design & Applications, 17(6), 2020, 1168-1176, https://doi.org/10.14733/cadaps.2020.1168-1176

[25] Tanaka, M.; Terano, M.; Higashino, C.; Asano, T.; Takasugi, K.: A Learning Method for Reconstructing 3D Models from Sketches, Computer-Aided Design & Applications, 16(6), 2019, 1158-1170, https://doi.org/10.14733/cadaps.2019.1158-1170

[26] Tatarchenko, M.; Richter, S.; Ranftl, R.; Li, Z.; Koltun, V.; Brox, T: What Do Single-View 3D Reconstruction Networks Learn?, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, 3405-3414, http://doi.org/10.1109/CVPR.2019.00352

[27] Varley, P. A. C.; Martin, R. R.; Suzuki, H.: Frontal geometry from sketches of engineering objects: is line labelling necessary?, Computer-Aided Design, 37(12), 2005, 1285–1307, https://doi.org/10.1016/j.cad.2005.01.002

[28] Yang, B.; Wen, H.; Wang, S.; Clark, R.; Markham, A.; Trigoni, N.: 3D Object Reconstruction from a Single Depth View with Adversarial Learning, Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, 679-688, http://doi.org/10.1109/ICCVW.2017.86

[29] Yang, B.; Rosa, S.; Markham, A.; Trigoni, N.; Wen, H.: Dense 3D Object Reconstruction from a Single Depth View, IEEE Transactions on Pattern Analysis and Machine Intelligence, 41(12), 2019, https://doi.org/10.1109/TPAMI.2018.2868195

[30] Zou, C.; Peng, X.; Lv, H.; Chen, S.; Fu, H.; Liu, J.: Sketch-based 3-D modeling for piecewise planar objects in single images, Computers & Graphics, 46, 2015, 130-137, http://doi.org/10.1016/j.cag.2014.09.031