www.cadanda.com Computer-Aided Design AND APPLICATIONS

A Learning Method for Reconstructing 3D Models from Sketches

Masaji Tanaka[1], Motoki Terano[1], Chiharu Higashino[2], Tetsuya Asano[2] and Keigo Takasugi[3]

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

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

ABSTRACT

Presently solid modelers have become popular tools in CAD/CAM systems, and three-dimensional (3D) sketching systems as simpler modelers have been developed in recent years. However, various basic operations are necessary to create solid models, making them difficult to use by beginners. Generally, sketches are convenient for expressing 3D models and most people can draw sketches. We have been developing a system that can automatically reconstruct 3D models from sketches that include curved lines. In this system, many of broken sketches be restored, so a learning method was introduced, but we could not explain its algorithm in detail. In this paper, an epochal learning method is proposed where a larger number of broken sketches be handled than by our previous method. Moreover, rough sketches can be corrected into precise sketches, and several examples are presented in this paper to demonstrate this method.

Keywords: Machine Learning, Reconstruction, Sketch, 3D Model, Line Drawing.
DOI: https://doi.org/10.14733/cadaps.2019.1158-1170

1 INTRODUCTION

Presently solid modelers have become popular tools in computer-aided design/manufacturing (CAD/CAM) systems. However, much study and training is required by a user to create a solid model of an object. Although many simpler solid modelers and 3D sketching systems have been developed in recent years, [9-10], a minimum number of operations must be performed in each of them. These operations tend to be unique to each system, so the systems are not standard solid modelers such as CATIA. As a result, most of them would likely not achieve significant popularity. We have been developing a system that can automatically reconstruct 3D models from 2D line sketches [18],[20]. Generally, sketches are convenient for expressing the shapes of 3D objects and most people can draw them without knowing formal CAD rules. Also, designers can easily discuss new parts, products, etc. with sketches. If our system is successful, it should become a powerful tool in CAD/CAM systems. In our original system, primitive sketches consisting of sketch features that are familiar to people are defined. Fig. 1 shows three samples. When a sketch is input to this system, a sketch feature is detected and extracted repeatedly until there are no lines remaining in the sketch. After each extraction of a sketch feature, remaining lines usually form a broken sketch. Therefore, it is necessary to restore them into a correct sketch for the next sketch feature extraction. However, as there are too many patterns in broken sketches, it is impossible to cover all of them through programming of the system. As a result, we introduced inductive learning techniques to learn restoration of broken sketches step by step. Although the learning technique was effective, we could not explain the detailed algorithm and its correction capability precisely.

In this paper, an epochal learning method is proposed in which each geometric element of a sketch is defined as an instance of a class. Also, each relationship between two geometric elements is defined as an instance of a class. So, many classes are first defined, such as Point class, Straight Line class, etc. in this method. Each class consists of many properties. Therefore, each geometric element and each relationship is expressed through the properties. For example, two endpoints, length, and direction are three properties of a straight line. When two or more examples are input, they can be generalized using this method. Each example consists of a question and answer (Q & A) formulation. For broken sketches, the question is a broken sketch and the answer is a restored sketch. Because these two sketches are expressed as properties, a means of restoration can be accomplished by changing the properties. Also, several properties can be changed into variables by inputting plural examples. Therefore, making variables is how examples are generalized in this method. Moreover, this generalization technique can be extended to the correction of rough sketches into precise sketches.

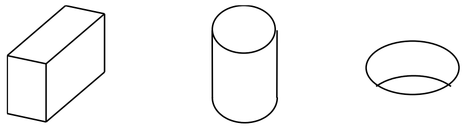


Figure 1: Three sketch features: (a) Cuboid, (b) Cylinder, and (c) Round Hole.

2 RELATED WORK

There are two types of related work concerning this paper. One involves reconstruction techniques of sketches into solid models. Another involves machine learning techniques concerned with CAD and computer graphics (CG). In the reconstruction techniques, their classification is well indicated in [3]. Originally, a line labeling technique was developed as Clowes-Huffman labeling, [2]. By labeling, using L, W, Y, and T junctions, straight lines were defined. Then many methods to automatically reconstruct solid models from sketches were developed. For example, Varley et al. attempted to provide depth reasoning of frontal geometry through sketches, [21-22]. Cao et al. [1] inferred hidden shapes using a unique method based on polyhedrons. Although these techniques were applied only to sketches consisting of straight lines, Malik [12] applied the line labeling technique to curved lines to handle sketches of 3D objects including curved faces. However, he did not mention how to create solid models from sketches that include curved lines. Although there are a few methods, [4-5] for handling objects including curved faces, they not generally apply to various types of curved objects.

Regarding learning techniques, basically our technique would be akin to inductive learning, [13]. In the technique, when plural Q & A are input as examples, they are generalized, so when a new question is input, its answer will be automatically output. Iwama et al. proposed an inductive learning technique for learning mathematical sentences, [8],[11]. The technique developed a program that uses procedures to check the truth or the falsity of expressions and sentences such as “7 is a prime number”. He developed generalization by using variables with similar input examples. We adopted his technique in our method. First we developed a method to restore partial omissions in 2D mechanical drawings, [16-17]. In the method, each geometric element such as straight lines and the relationships of geometric elements were defined as properties. Restoration of the omissions was learned as the change of properties. The more restoration patterns used for learning, the more properties would be required, which was an issue. So, we developed a method referred to as Inductive Functional prOgramming for Geometric processing (IFOG) to minimize the number of properties, [19]. In IFOG, each geometric element was defined as a class consisting of properties. Therefore, a sketch consisted of the instances of classes, and whenever some properties were necessary, they were generated from instances to classes by the method. Next, a learning method to restore broken sketches was proposed. However, we could not explain the algorithm in detail.

In recent years, a great many machine learning systems have been proposed and developed. Also, for developing advanced CAD systems, many researchers have applied machine learning techniques such as deep learning and Genetic Algorithms (GAs), [6][14]. Especially, deep learning has become popular and its applications are increasing drastically, [15]. However, it is not clear how a deep learning method constructs a procedure out of example sequences of steps unless the method produces each step out of example steps and some other method can map the steps. Also, as deep learning methods are based on traditional neural network techniques, their organization is complex and very large amounts of data are required. In our proposed learning method in this paper, a procedure can be constructed from example sequences of steps. Also, this method can learn a procedure from only a few examples, and its organization is simpler and clearer than other learning techniques.

3 ALGORITHM FOR RECONSTRUCTING 3D MODELS FROM SKETCHES

For clarity, our sketches in this paper are limited as follows. Each sketch is drawn using straight lines, ellipses, and elliptical arcs in 2D CAD systems. Also, each sketch is drawn from a general view correctly and precisely, and hidden lines are not drawn. In this section, the algorithm for reconstructing 3D models from sketches in our system is explained with Example 1 illustrated in Fig. 2(a). A more detailed explanation of the algorithm is presented in [18].

(1) When Example 1 is input, all straight lines are divided at their intersections except curves. At each L-junction or W-junction, lines are extended only within the sketch. Each extended part is recognized as an additional line. In Example 1, there are two W-junctions and an L-junction ($W1$, $W2$, $L1$), so six additional lines can be drawn as dotted lines as shown in Fig. 2(b).

(2) A sketch feature is detected and extracted. It can be converted to a 3D feature with the “cubic corner method”, [21]. Fig. 2(c) shows the detection of a cuboid sketch, and Fig. 2(d) shows its extraction as $f1$.

(3) After each feature extraction, several isolated lines (each of which does not involve a closed loop) are often generated. In Fig. 2(d), there is an isolated and dotted straight line. When the line is extended upward, its terminal never makes contact with any other lines. Therefore, it can be removed, as shown in Fig. 2(e).

(4) The feature extraction is continued unless there are no lines remaining in the input sketch. In Fig. 2(e), the other cuboid sketch ($f2$) can be detected, and extracted, as shown in Fig. 2(f). There are many isolated lines in this figure. When isolated lines are removed step by step, Fig. 2(g) is obtained. In this figure, the sketch of a round hole can be restored as $f3$. After the extraction of $f3$ and some restoration process, $f4$ can be detected, as shown in Fig. 2(h). For restoration of $f3$ and $f4$, our learning technique can be applied.

(5) All 3D features are combined in accordance with the input sketch inductively, and the solution is obtained. In Example 1, a solution obtained is $f4 + f3 + f2 + f1$, as shown in Fig. 2(i).

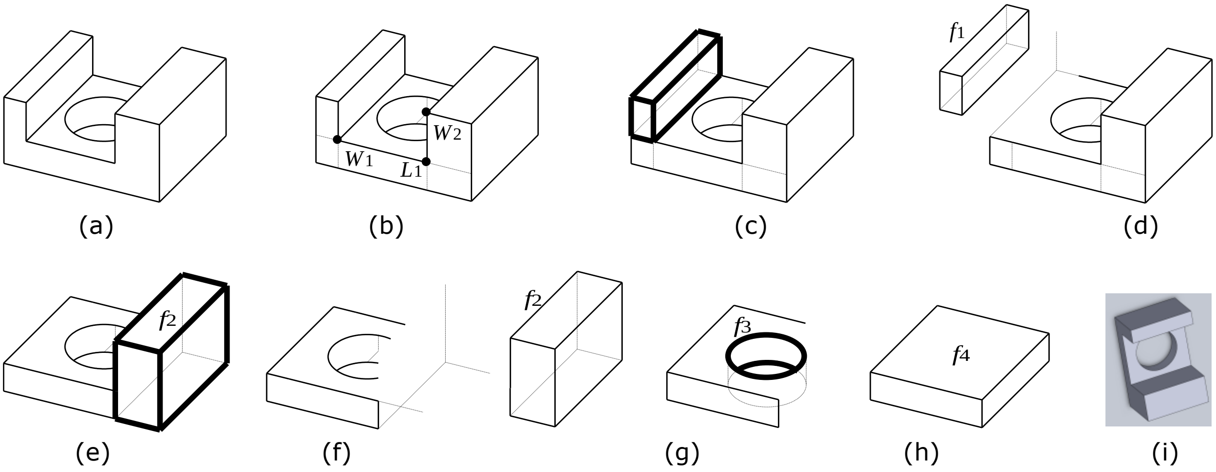


Figure 2: Example 1 and the process to arrive at the solution: (a) Example 1, (b) Generation of additional lines, (c) Detection of a cuboid sketch, (d) Extraction of *f*1, (e) Detection of *f*2, (f) Extraction of *f*2, (g) Detection of *f*3, (h) Detection of *f*4, and (i) The solution.

4 LEARNING METHOD FOR THE RESTORATION OF BROKEN SKETCHES

4.1 Main Algorithm

Fig. 3 shows the algorithm for our proposed learning method. All geometric data is classified into six classes as Point, Straight line, Elliptical arc, Relation between two straight lines, Relation between two elliptical arcs, and Relation between elliptical arc and straight line (see Appendix A). Each class consists of properties. For example, the Point class consists of five properties. Here, x-y coordinates are ignored in the Point class because they are inherent and redundant. Fig. 3 shows the algorithm for this method. Each step is explained with Example 2 illustrated in Fig. 4 and Example 3 illustrated in Fig. 5 as follows.

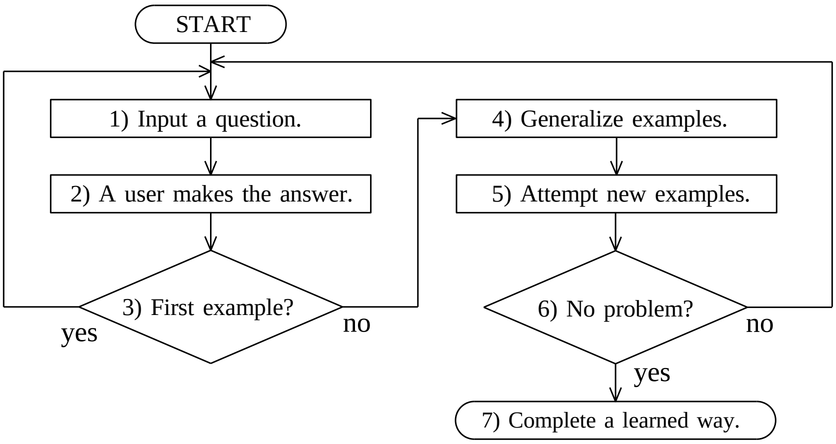
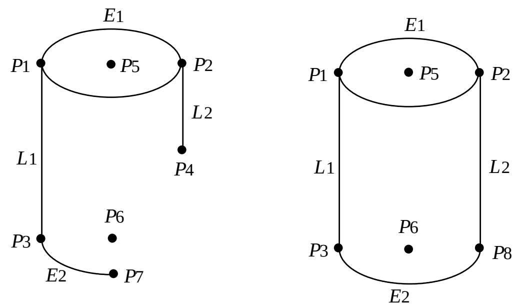
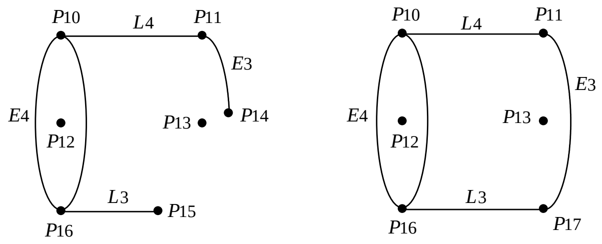


Figure 3: Algorithm for our proposed learning method.

Fig. 4 shows Example 2. Fig. 4(a) shows a broken sketch of a cylinder, and Fig. 4(b) shows a restored sketch of Fig. 4(a). These two figures correspond to a single Q & A. So, first Fig. 4(a) is input to our method as a question in Step 1 of Fig. 3. In Step 2, Fig. 4(b) is drawn from Fig. 4(a) by a user directly with a 2D CAD system. In Step 3, as this Q & A is the first example for learning a restoration for broken cylinders, Step 1 is executed again. Similarly, Example 3 illustrated in Fig. 5 is input and then Step 4 is executed. In this step, these two examples can be generalized. This generalization process is explained in detail as follows. First Example 2 can be expressed as properties, as shown in Appendix B1.



(a) (b)



(a) (b)

Figure 4: Example 2: (a) A broken cylinder, cylinder, and (b) Restored.

Figure 5: Example 3: (a) Another broken and (b) Restored.

For example, the length of *L*2 is 24.23 in Fig. 4(a) but it is changed to 50 in Fig. 4(b). Also, there are three relations between elliptical arcs and straight lines contacting each other in Fig. 4(a) but there are four relations between them in Fig. 4(b). Therefore, this restoration process can be expressed as a change of their properties. In the same manner, Example 3 can be expressed as properties as shown in Appendix B2. When two questions (Fig. 4(a), Fig. 5(a)) are compared, there are many similar points in their properties. So, it is difficult to generalize them initially. Therefore, straight lines are first generalized. For example, the properties of *L*1 and *L*4 are the same at 5) and 6) but *L*1 and *L*2 are different. Therefore, *L*1, *L*4 and *L*2, *L*3 have the same role respectively in the two broken cylinders. As a result, they can be generalized as *L*x1 and *L*x2 respectively. *L*x1 and *L*x2 are two variables of straight lines. Similarly *E*1, *E*4 and *E*2, *E*3 can be generalized as *E*x1, *E*x2 respectively. When these four variables are applied to points, they can be generalized as variable points. The correspondence of all variables indicated above is as follows.

Lx1 = {L1, L4}, Lx2 = {L2, L3}, Ex1 = {E1, E4}, Ex2 = {E2, E3}, Px1 = {P1, P10}, Px2 = {P2, P16}, Px3 = {P3, P11},
Px4 = {P4, P15}, Px5 = {P5, P12}, Px6 = {P6, P13}, Px7 = {P7, P14}.

They can satisfy all relations among straight lines and elliptical arcs in the two questions. Next, two answers (Fig. 4(b), Fig. 5(b)) are compared. Obviously *P*x4 and *P*x7 are removed by the generalization of *P*x8 (={*P*8, *P*17}). As a result, the two answers can be generalized. Finally, the two examples can be generalized as shown in Appendix B3. In Step 5, a new question is input. Fig. 6 shows an example. When this question is applied to the generalized question, all variables can be changed into real values step by step (Appendix B4).

For example, if the length of *L*1 in Fig. 6(b) is incorrect, some example indicating that the lengths of two straight lines are different from the other examples has to be input in Step 1. As there are no problems in Example 4, a restoration for broken cylinders is completely learned in Step 7.

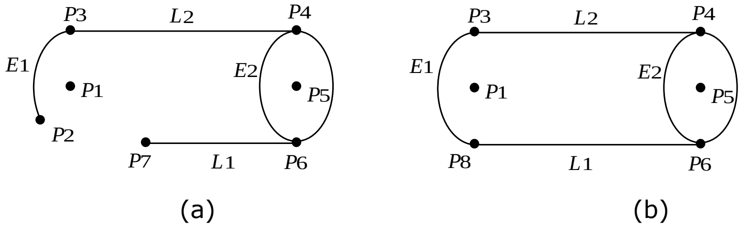


Figure 6: Example 4: (a) A new broken cylinder, and (b) Restored (a).

4.2 Applications for Restoring Cylinder Sketches

Fig. 7 shows applications for restoring cylinder sketches described above. Fig. 7(a) shows a broken cylinder. Because its broken part is different from Fig. 4(a), new learning must take place required for restoring this figure. In the case of Fig. 7(b), the total length of the cylinder is unknown. In this case, a user can decide the length (such as 100) while he/she inputs two or more examples. This technique can be applied to Fig. 7(c) that has eight identical cylinders. When the value of the property of "4) Angle" in the Class Relation between Two Straight Lines is not zero, tapers such as Fig. 7(d) can be handled by this method. In Fig. 7(e) (sketch of a pipe), two straight lines are changed to two curves, and in Fig. 7(f), those two lines do not exist and a sketch of a round hole is drawn. They can be handled as applications for restoring cylinders by this method. Moreover, in Example 1, $f3$ and $f4$ can also be restored clearly by this method.

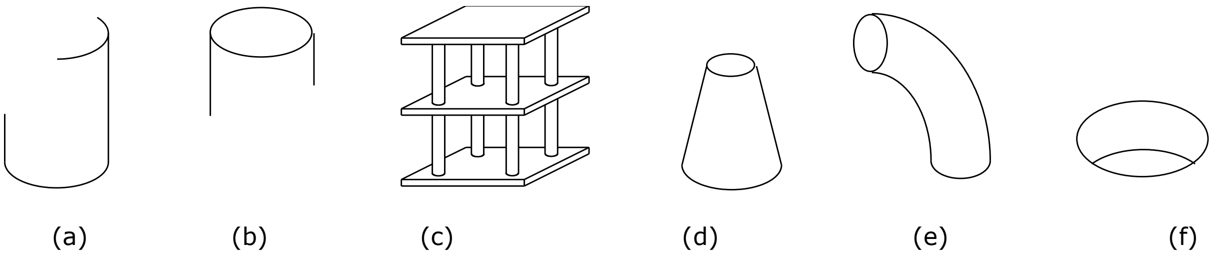


Figure 7: Applications for restoring cylinder sketches: (a) Another broken cylinder, (b) Half broken cylinder, (c) Application sketch of (b), (d) Sketch of tapered cylinder, (e) Sketch of a pipe, and (f) Sketch of a round hole.

5 LEARNING METHOD FOR THE CORRECTION OF ROUGH SKETCHES

When a user draws a sketch on a PC, tablet, etc., it will be rough and not be precise. Our proposed method can be applied to correct rough sketches into precise sketches because rough sketches are similar to broken sketches. In this section, the way to correct rough Y-junctions is explained. Next, the way to correct the rough sketches of cuboids is explained. Finally, an application is presented.

5.1 Correction of Rough Y-junctions

Fig. 8(a) shows a rough sketch of a Y-junction. In our method, this figure becomes a question. Here, although free hand lines are more useful, how to handle them is a very difficult issue [7]. Therefore, this issue is ignored in this paper. For this question, Point class and Straight line class are updated as shown in Appendix C1. Especially, vertical lines are important for applying the "cubic corner method", [21]. Therefore, it is necessary for users to decide the allowable range of verticality. In Fig. 8(b), $L3$ is corrected to vertical. Then $L1$ and $L3$ make contact by extending them.

This figure represents a middle answer of the question. In Fig. 8(c), $L2$ is extended to $P7$. This figure becomes an answer to the question. The data of instances in these figures are indicated in Appendix C2. When another Q & A is input, they can be generalized, as shown in Appendix C3. When this generalization is applied to Fig. 9(a) as a new question, Fig. 9(b) can be obtained automatically.

5.2 Correction of Rough Cuboids and Its Applications

In the same way, rough L-junctions and W-junctions can be corrected automatically. Fig. 10(a) shows a rough sketch of a cuboid. When all junctions are corrected with this method, Fig. 10(b) can be obtained automatically. The cuboid does not consist of three correct parallelograms. In this case, three straight lines ($L1$, $L2$, $L3$) form a Y-junction ($P1$), so the other three lines can follow them for learning correct cuboid sketches. As a result, a correct sketch of the cuboid can be obtained, as shown in Fig. 10(c).

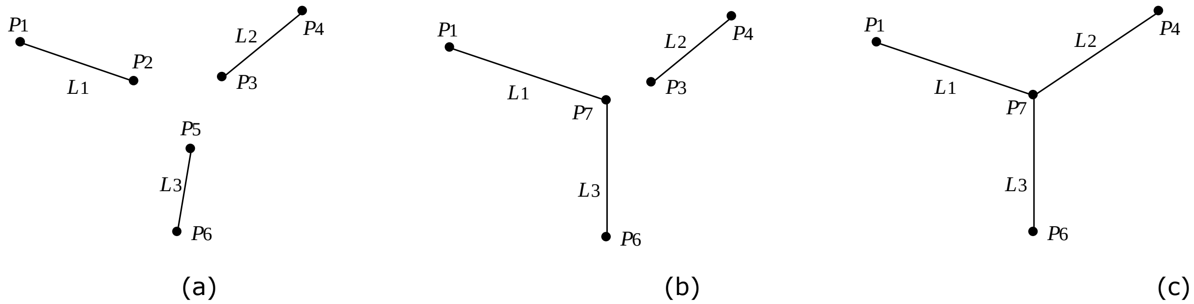


Figure 8: Correction of a Y-junction: (a) Question, (b) Middle answer, and (c) Final answer.

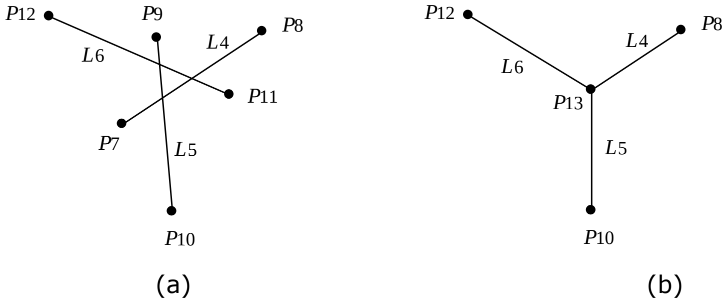


Figure 9: New question of a Y-junction: (a) Question, and (b) Answer.

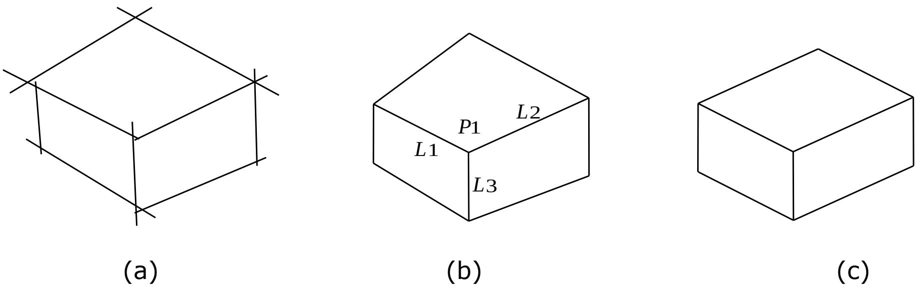


Figure 10: Correction of a rough sketch of a cuboid: (a) Rough sketch of cuboid, (b) Correction of junctions, and (c) Fully corrected.

The correction processes in this section are effective for rough sketches of sketch features, but it would be difficult to handle rough sketches of complex objects. Therefore, the authors have been developing a sketch feature based solid modeler as follows. This solid modeler makes solid models from rough sketches of sketch features step by step. In Section 2, our system creates a solid model by extracting sketch features step by step from an input sketch such as Example 1. Fig. 11 shows an example to make a solid model from the solution of Example 1. First, a rough sketch of a cuboid is input as in Fig. 11(a). In Fig. 11(b), this cuboid is corrected and its solid model can be obtained. Also, a rough sketch of a round hole is drawn in this figure. In Fig. 11(c), the hole is corrected and a 3D hole is added to the 3D cuboid. In Fig. 11(d), a rough sketch of the other cuboid is drawn using two straight line segments of the existing cuboid. In Fig. 11(e), the two cuboids are combined, and the third rough sketch of a cuboid is drawn. In Fig. 11(f), this cuboid is corrected and a solid model is completed.

If more sketch features are applied, more complex 3D models such as Example 5, illustrated in Fig. 12 can be constructed from the modeler. In Example 5, first a user draws two disks as in Fig. 12(a). Then the modeler can convert the sketch to a 3D model as in Fig. 12(b). Next, the user draws a pole between the disks as in Fig. 12(c). So the 3D model of this figure is updated as in Fig. 12(d). Because this figure shows a solid model, it can be rotated and another pole can be drawn as in Fig. 12(e). Then this figure can be converted to a 3D model as in Fig. 12(f). When many poles are drawn, a solid model of an accessory case can be obtained, such as in Fig. 12(f).

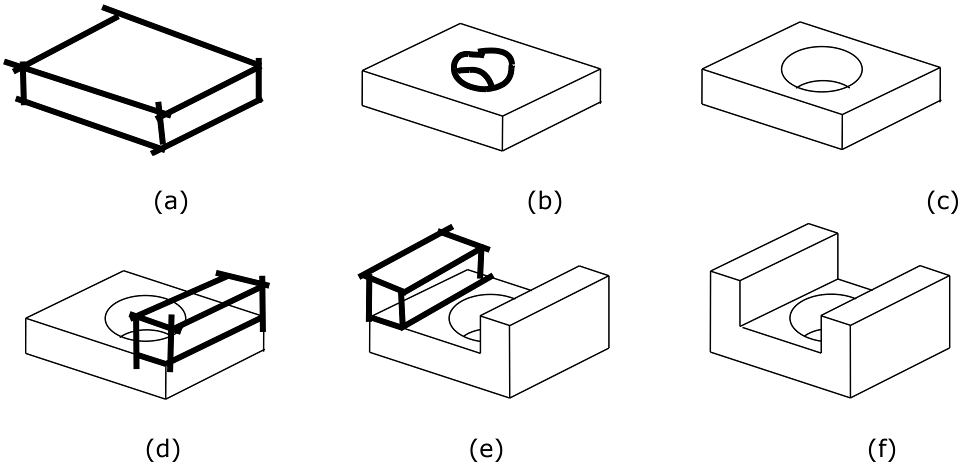
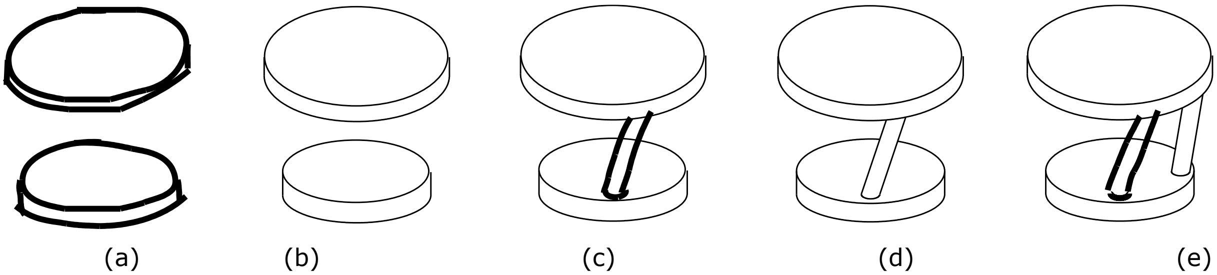
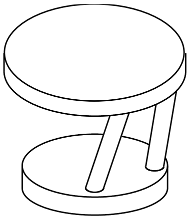
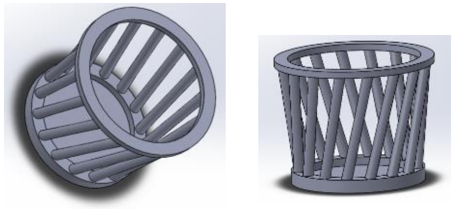


Figure 11: Creating of a solid model of Example 1 from rough sketch features: (a) Rough sketch of cuboid, (b) Correction of (a) and a rough sketch of round hole, (c) Corrected hole, (d) Rough sketch of second cuboid, (e) Corrected (d) and rough sketch of third cuboid, and (f) Completion of the solid model.





(f)



(g)

Figure 12: Example 5: (a) Rough sketch of two disks, (b) 3D model of (a), (c) Addition of a pole sketch, (d) 3D model of (c), (e) Addition of another pole sketch, (f) 3D model of (f), and (g) Two overviews of the solid model of an accessory case.

6 CONCLUSIONS

In this paper, our proposed learning method is applied to restore broken sketches to reconstruct 3D models from sketches. However, applicable sketches would be quite limited in our reconstruction system because most sketches of objects are not simple assemblies of sketch features. Therefore, we show how this method can automatically correct rough sketches into precise sketches, using a new type of solid modeler. In this solid modeler, solid models are made from assembling 3D sketch features, so applicable 3D objects can be created using step-by-step rough sketches using our reconstruction system. The effectiveness of this method is apparent for either complete sketches or to develop full 3D models from user sketches.

Appendix A

Class Point:
1) Number, 2) {Contact lines}, 3) Number of 2), 4) Center point(s) of elliptical arcs?, 5) Tangent point of lines?;
Class Straight Line:
1) Number, 2) {Two endpoints}, 3) Length, 4) Direction (0<=θ<180), 5) {Isolated points}, 6) Number of 5);
Class Elliptical Arc:
1) Number, 2) Length of long axis, 3) Length of short axis, 4) {Two endpoints}, 5) Center point, 6) {Isolated endpoints}, 7) Number of 6), 8) Ellipse?, 9) Direction of 3) (0<=θ<180);
Class Relation between Two Straight Lines:
1) {Two lines}, 2) Longer line, 3) More isolated line, 4) Angle (0<=θ<180), 5) Contact point;
Class Relation between Two Elliptical Arcs:
1) {Two arcs}, 2) Wider arc, 3) More isolated arc, 4) {Contact points}, 5) Number of 4), 6) If their directions are the same, what is the direction?;
Class Relation between Contacted Elliptical Arc and Straight Line:
1)Elliptical Arc, 2) Straight Line, 3) Contact point, 4) Is 3) a tangent point?, 5) Is 1) an ellipse?;

Appendix B1

Instances in Fig. 4(a):
(Point) 1) P1 2) {L1, E1} 3) 2 4) NA 5) yes; 1) P2 2) {L2, E1} 3) 2 4) NA 5) yes; 1) P3 2) {L1, E2} 3) 2 4) NA 5) yes;
1) P4 2) {L2} 3) 1 4) NA 5) no; 1) P5 2) {φ} 3) 0 4) E1 5) no; 1) P6 2) {φ} 3) 0 4) E2 5) no; 1) P7 2) {E2} 3) 1 4) NA 5) no;
(Straight Line) 1) L1 2) {P1, P3} 3) 50.0 4) 90.0deg 5) {φ} 6) 0; 1) L2 2) {P2, P4} 3) 24.23 4) 90.0deg 5) {P4} 6) 1;
(Elliptical Arc) 1) E1 2) 40.0 3) 20.0 4) {φ} 5) P5 6) {φ} 7) 0 8) yes 9) 90.0deg;

1) E2 2) 40.0 3) 20.0 4) {P3, P7} 5) P6 6) {P7} 7) 1 8) no 9) 90.0deg;
(Relation between Two Straight Lines) 1) {L1, L2} 2) L1 3) L2 4) 0.0deg 5) NA;
(Relation between Two Elliptical Arcs) 1) {E1, E2} 2) NA 3) E2 4) {φ} 5) 0 6) 90.0deg;
(Relation between Contacted Elliptical Arc and Straight Line)
1) E1 2) L1 3) P1 4) yes 5) yes; 1) E1 2) L2 3) P2 4) yes 5) yes; 1) E2 2) L1 3) P3 4) yes 5) no;

Instances in Fig. 4(b):
(Point) 1) P1 2) {L1, E1} 3) 2 4) NA 5) yes; 1) P2 2) {L2, E1} 3) 2 4) NA 5) yes; 1) P3 2) {L1, E2} 3) 2 4) NA 5) yes;
1) P5 2) {φ} 3) 0 4) E1 5) no; 1) P6 2) {φ} 3) 0 4) E2 5) no; 1) P8 2) {L2, E2} 3) 2 4) NA 5) yes;
(Straight Line) 1) L1 2) {P1, P3} 3) 50.0 4) 90.0deg 5) {φ} 6) 0; 1) L2 2) {P2, P8} 3) 50.0 4) 90.0deg 5) {φ} 6) 0;
(Elliptical Arc) 1) E1 2) 40.0 3) 20.0 4) {φ} 5) P5 6) {φ} 7) 0 8) yes 9) 90.0deg;
1) E2 2) 40.0 3) 20.0 4) {P3, P8} 5) P6 6) {φ} 7) 0 8) no 9) 90.0deg;
(Relation between Two Straight Lines) 1) {L1, L2} 2) NA 3) NA 4) 0.0deg 5) NA;
(Relation between Two Elliptical Arcs) 1) {E1, E2} 2) NA 3) NA 4) {φ} 5) 0 6) 90.0deg;
(Relation between Contacted Elliptical Arc and Straight Line)
1)E1 2)L1 3)P1 4) yes 5) yes; 1)E1 2)L2 3)P2 4) yes 5) yes; 1)E2 2)L1 3)P3 4) yes 5) no; 1)E2 2)L2 3)P8 4) yes 5) no;

Appendix B2

Instances in Fig. 5(a):
(Point) 1) P10 2) {L4, E4} 3) 2 4) NA 5) yes; 1) P11 2) {L4, E3} 3) 2 4) NA 5) yes; 1) P12 2) {φ} 3) 0 4) E4 5) no;
1) P13 2) {φ} 3) 0 4) E3 5) no; 1) P14 2) {E3} 3) 1 4) NA 5) no; 1) P15 2) {L3} 3) 1 4) NA 5) no;
1) P16 2) {E4, L3} 3) 2 4) NA 5) yes;
(Straight Line) 1) L4 2) {P10, P11} 3) 35.0 4) 0.0deg 5) {φ} 6) 0; 1) L3 2) {P15, P16} 3) 24.5 4) 0.0deg 5) {P15} 6) 1;
(Elliptical Arc) 1) E4 2) 60.0 3) 15.0 4) {φ} 5) P12 6) {φ} 7) 0 8) yes 9) 0.0deg;
1) E3 2) 60.0 3) 15.0 4) {P11, P14} 5) P13 6) {P14} 7) 1 8) no 9) 0.0deg;
(Relation between Two Straight Lines) 1) {L4, L3} 2) L4 3) L3 4) 0.0deg 5) NA;
(Relation between Two Elliptical Arcs) 1) {E4, E3} 2) NA 3) E3 4) {φ} 5) 0 6) 0.0deg;
(Relation between Contacted Elliptical Arc and Straight Line)
1) E4 2) L4 3) P10 4) yes 5) yes; 1) E4 2) L3 3) P16 4) yes 5) yes; 1) E3 2) L4 3) P11 4) yes 5) no;

Instances in Fig. 5(b):
(Point) 1) P10 2) {L4, E4} 3) 2 4) NA 5) yes; 1) P11 2) {L4, E3} 3) 2 4) NA 5) yes; 1) P12 2) {φ} 3) 0 4) E4 5) no;
1) P13 2) {φ} 3) 0 4) E3 5) no; 1) P16 2) {E4, L3} 3) 2 4) NA 5) yes; 1) P17 2) {E3, L3} 3) 2 4) NA 5) yes;
(Straight Line) 1) L4 2) {P10, P11} 3) 35.0 4) 0.0deg 5) {φ} 6) 0; 1) L3 2) {P17, P16} 3) 35.0 4) 0.0deg 5) {φ} 6) 0;
(Elliptical Arc) 1) E4 2) 60.0 3) 15.0 4) {φ} 5) P12 6) {φ} 7) 0 8) yes 9) 0.0deg;
1) E3 2) 60.0 3) 15.0 4) {P11, P17} 5) P13 6) {φ} 7) 0 8) no 9) 0.0deg;
(Relation between Two Straight Lines) 1) {L4, L3} 2) NA 3) NA 4) 0.0deg 5) NA;
(Relation between Two Elliptical Arcs) 1) {E4, E3} 2) NA 3) NA 4) {φ} 5) 0 6) 0.0deg;
(Relation between Contacted Elliptical Arc and Straight Line)
1)E4 2)L4 3)P10 4) yes 5) yes; 1)E4 2)L3 3)P16 4) yes 5) yes; 1)E3 2)L4 3)P11 4) yes 5) no; 1)E3 2)L3 3)P17 4) yes 5) no;

Appendix B3

Here, $x1$, $x2$, $x3$, and $x4$ are variables of real numbers. ”deg1” is a variable of an angle.

Generalized question for a broken cylinder:
(Point) 1)Px1 2){Lx1, Ex1} 3)2 4)NA 5)yes; 1)Px2 2){Lx2, Ex1} 3)2 4)NA 5)yes; 1)Px3 2){Lx1, Ex2} 3)2 4)NA 5)yes;

1)Px4 2){Lx2} 3)1 4)NA 5)no; 1)Px5 2){φ} 3)0 4)Ex1 5)no; 1)Px6 2){φ} 3)0 4)Ex2 5)no; 1)Px7 2){Ex2} 3)1 4)NA 5)no;
(Straight Line) 1) Lx1 2) {Px1, Px3} 3) x1 4) deg1 5) {φ} 6) 0; 1) Lx2 2) {Px2, Px4} 3) x2 4) deg1 5) {Px4} 6) 1;
(Elliptical Arc) 1) Ex1 2) x3 3) x4 4) {φ} 5) Px5 6) {φ} 7) 0 8) yes 9) deg1;
1) Ex2 2) x3 3) x4 4) {Px3, Px7} 5) Px6 6) {Px7} 7) 1 8) no 9) deg1;
(Relation between Two Straight Lines) 1) {Lx1, Lx2} 2) Lx1 3) Lx2 4) 0.0deg 5) NA;
(Relation between Two Elliptical Arcs) 1) {Ex1, Ex2} 2) NA 3) Ex2 4) {φ} 5) 0 6) deg1;
(Relation between Contacted Elliptical Arc and Straight Line)
1)Ex1 2)Lx1 3)Px1 4)yes 5)yes; 1)Ex1 2)Lx2 3)Px2 4)yes 5)yes; 1)Ex2 2)Lx1 3)Px3 4)yes 5)no;

Generalized answer for a broken cylinder:
(Point) 1)Px1 2){Lx1, Ex1} 3)2 4)NA 5)yes; 1)Px2 2){Lx2, Ex1} 3)2 4)NA 5)yes; 1)Px3 2){Lx1, Ex2} 3)2 4)NA 5)yes;
1)Px5 2){φ} 3)0 4)Ex1 5)no; 1) Px6 2){φ} 3)0 4)Ex2 5)no; 1)Px8 2){Lx2, Ex2} 3)2 4)NA 5)yes;
(Straight Line) 1) Lx1 2) {Px1, Px3} 3) x1 4) deg1 5) {φ} 6) 0; 1) Lx2 2) {Px2, Px8} 3) x1 4) deg1 5) {φ} 6) 0;
(Elliptical Arc) 1) Ex1 2) x3 3) x4 4) {φ} 5) Px5 6) {φ} 7) 0 8) yes 9) deg1;
1) Ex2 2) x3 3) x4 4) {Px3, Px8} 5) Px6 6) {φ} 7) 0 8) no 9) deg1;
(Relation between Two Straight Lines) 1) {Lx1, Lx2} 2) NA 3) NA 4) 0.0deg 5) NA;
(Relation between Two Elliptical Arcs) 1) {Ex1, Ex2} 2) NA 3) NA 4) {φ} 5) 0 6) deg1;
(Relation between Contacted Elliptical Arc and Straight Line)
1)Ex1 2)Lx1 3)Px1 4)yes 5)yes; 1)Ex1 2)Lx2 3)Px2 4)yes 5)yes; 1)Ex2 2)Lx1 3)Px3 4)yes 5)no;
1)Ex2 2)Lx2 3)Px8 4)yes 5)no;

Appendix B4

First, three relations can be changed as follows.
(Relation between Two Straight Lines) 1) {L1, L2} 2) L2 3) L1 4) 0.0deg 5) NA;
(Relation between Two Elliptical Arcs) 1) {E1, E2} 2) NA 3) E1 4) {φ} 5) 0 6) 0.0deg;
(Relation between Contacted Elliptical Arc and Straight Line)
1)E2 2)L1 3)P6 4)yes 5)yes; 1)E2 2)L2 3)P4 4)yes 5)yes; 1)E1 2)L2 3)P3 4)yes 5)no;
Here, $Lx1 = L1$, $Lx2 = L2$, $Ex1 = E2$, and $Ex2 = E1$. Then all remaining geometric elements can be specified. Continuously, a real answer can be obtained from the generalized answer. As a result, Fig. 6(b) is obtained automatically.

Appendix C1

Here, "6)" is added in the Point class, and "7), 8), 9)" are added in the Straight Line class.

Updated Class Point:
1) Number, 2) {Contact lines}, 3) Number of 2), 4) Center point(s) of elliptical arcs?, 5) Tangent point of lines?, 6) Y-junction point?;
Updated Class Straight Line:
1) Number, 2) {Two endpoints}, 3) Length, 4) Direction (0<=θ<180), 5) {Isolated points}, 6) Number of 5), 7) About vertical?, 8) If 7) is yes, which is upper endpoint?, 9) If 7) is no, which is lower?;

Appendix C2

Instances in Fig. 8(a):
(Point) 1) P1 2) {L1} 3) 1 4) NA 5) no 6) no; 1) P2 2) {L1} 3) 1 4) NA 5) no 6) no; 1) P3 2) {L2} 3) 1 4) NA 5) no 6) no; 1) P4 2) {L2} 3) 1 4) NA 5) no 6) no; 1) P5 2) {L3} 3) 1 4) NA 5) no 6) no; 1) P6 2) {L3} 3) 1 4) NA 5) no 6) no;
(Straight Line) 1) L1 2) {P1, P2} 3) 34.5 4) 167.7deg 5) {P1, P2} 6) 2 7) no 8) P1 9) NA; 1) L2 2) {P3, P4} 3) 29.3 4) 45.1deg 5) {P3, P4} 6) 2 7) no 8) P4 9) NA; 1) L3 2) {P5, P6} 3) 31.2 4) 86.4deg 5) {P5, P6} 6) 2 7) yes 8) NA 9) P6;

(Relation between Two Straight Lines)
1){L1, L2} 2)L1 3)NA 4)122.6deg 5)NA; 1){L1, L3} 2)L1 3)NA 4)98.7deg 5)NA; 1){L2, L3} 2)L3 3)NA 4)138.7deg 5)NA;

Instances in Fig. 8(c):
(Point) 1) P1 2) {L1} 3) 1 4) NA 5) no 6) no; 1) P4 2) {L2} 3) 1 4) NA 5) no 6) no; 1) P6 2) {L3} 3) 1 4) NA 5) no 6) no; 1) P7 2) {L1, L2, L3} 3) 3 4) NA 5) no 6) yes;
(Straight Line) 1) L1 2) {P1, P7} 3) 42.7 4) 167.7deg 5) {P1} 6) 1 7) no 8) P1 9) NA; 1) L2 2) {P7, P4} 3) 40.6 4) 40.0deg 5) {P4} 6) 1 7) no 8) P4 9) NA; 1) L3 2) {P7, P6} 3) 35.5 4) 90.0deg 5) {P6} 6) 1 7) yes 8) NA 9) P6;
(Relation between Two Straight Lines) 1) {L1, L2} 2) L1 3) NA 4) 127.7deg 5) P7; 1) {L1, L3} 2) L1 3) NA 4) 102.3deg 5) P7; 1) {L2, L3} 2) L2 3) NA 4) 130.0deg 5) P7;

Appendix C3

Here, *L*x1 and *L*x2 can be swapped.

Generalized question for Y-junction:
(Point) 1)Px1 2){Lx1} 3)1 4)NA 5)no 6)no; 1)Px4 2){Lx1} 3)1 4)NA 5)no 6)no; 1)Px5 2){Lx2} 3)1 4)NA 5)no 6)no;
1)Px2 2){Lx2} 3)1 4)NA 5)no 6)no; 1)Px3 2){Lx3} 3)1 4)NA 5)no 6)no; 1)Px6 2){Lx3} 3)1 4)NA 5)no 6)no;
(Straight Line) 1) Lx1 2) {Px1, Px4} 3) x1 4) deg1 5) {Px1, Px4} 6) 2 7) no 8) Px1 9) NA; 1) Lx2 2) {Px5, Px2} 3) x2 4) deg2 5) {Px5, Px2} 6) 2 7) no 8) Px2 9) NA; 1) Lx3 2) {Px3, Px6} 3) x3 4) deg3 5) {Px3, Px6} 6) 2 7) yes 8) NA 9) Px6;
(Relation between Two Straight Lines)
1){Lx1, Lx2} 2)Lx4 3)NA 4)deg4 5)NA; 1){Lx1, Lx3} 2)Lx5 3)NA 4)deg5 5)NA; 1){Lx2, Lx3} 2)Lx6 3)NA 4)deg6 5)NA;

Generalized answer for Y-junction:
(Point) 1) Px1 2) {Lx1} 3) 1 4) NA 5) no 6) no; 1) Px2 2) {Lx2} 3) 1 4) NA 5) no 6) no; 1) Px6 2) {Lx3} 3) 1 4) NA 5) no 6) no; 1) Px7 2) {Lx1, Lx2, Lx3} 3) 3 4) NA 5) no 6) yes;
(Straight Line) 1) Lx1 2) {Px1, Px7} 3) x4 4) deg1 5) {Px1} 6) 1 7) no 8) Px1 9) NA; 1) Lx2 2) {Px2, Px7} 3) x5 4) deg7 5) {Px2} 6) 1 7) no 8) Px2 9) NA; 1) Lx3 2) {Px6, Px7} 3) x6 4) 90.0deg 5) {Px6} 6) 1 7) yes 8) NA 9) Px6;
(Relation between Two Straight Lines) 1) {Lx1, Lx2} 2) Lx7 3) NA 4) deg8 5) Px7; 1) {Lx1, Lx3} 2) Lx8 3) NA 4) deg9 5) Px7; 1) {Lx2, Lx3} 2) Lx9 3) NA 4) deg10 5) Px7;

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
Keigo Takasugi, http://orcid.org/0000-0002-6080-7856

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] Clowes, M. B: On seeing things, Artificial Intelligence, 2(1), 1971, 79–116, http://dx.doi.org/10.1016/0004-3702(71)90005-1
[3] 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
[4] 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.

[5] 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

[6] Cui, H.; Turan, O.; Sayer, P.: Learning-based ship design optimization approach, Computer-Aided Design, 44(3), 2012, 186-195, http://dx.doi.org/10.1016/j.cad.2011.06.011

[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] Fujiwara, M.; Iwama, K.: A Program that acquires how to execute sentences, Wseas Trans. on Computers, 8, 2009, 1348–1357.

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

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

[11] Iwama, K.: A robotic program that acquires concepts and begins introspection, NeuroQuantology, 4, 2006, 321–328.

[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] Muggleton, S.; De Raedt, L.: Inductive logic programming: Theory and methods, 19(20), 1994, 629-679, http://dx.doi.org/10.1016/0743-1066(94)90035-3

[14] Qin, F.; Li, L.; Gao, S.; Yang, X.; Chen, X.: A deep learning approach to the classification of 3D CAD models, Journal of Zhejiang University SCIENCE C, 15(2), 2014, 91-106, http://dx.doi.org /10.1631/jzus.C1300185

[15] Schmidhuber, J.: Deep learning in neural networks: An overview, Neural Networks, 61, 2015, 85-117, http://dx.doi.org/10.1016/j.neunet.2014.09.003

[16] Tanaka, M; Kaneeda, T.; Yamahira, T.; Iwama, K.: A method to restore partial omissions in 2D drawings, Computer-Aided Design & Applications, 3, 2006, 341-347, http://dx.doi.org/10.1080/16864360.2006.10738472

[17] Tanaka, M.; Kaneeda, T.; Sasae, D.; Fukagawa, J.; Yokoi, R.: The Learning System to Restore Operations of Isolated Line Segments in 2D Drawings, Computer-Aided Design & Applications, 5(1-4), 2008, 354-362, http://dx.doi.org/10.3722/cadaps.2008.354-362

[18] 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

[19] Tanaka, M.; Takamiya, Y.; Tsubota, N.; Iwama, K: IFOG: Inductive Functional Programming for Geometric Processing, Computer-Aided Design & Applications, 13(3), 2015.

[20] 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

[21] 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.

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