COMPUTER-AIDED DESIGN & APPLICATIONS, 2017
VOL. 14, NO. 5, 632–641
http://dx.doi.org/10.1080/16864360.2016.1273580

Reconstruction of solid models from sketches including curves with inductive learning technique

Masaji Tanaka[a], Yuki Takamiya[a], Naoki Tsubota[a], Shigeyuki Asanuma[a] and Kenzo Iwama[b]

[a]Okayama University of Science, Japan; [b]EngiCom Corporation, Japan

ABSTRACT
Since decades, many methods have been proposed to automatically reconstruct solid models from their sketches. However, it was difficult for the methods to handle sketches including curved lines such as cylinders. Recent years, the authors have proposed a method for the reconstruction of sketches including curves. In the method, the features of sketches of cuboids, cylinders, etc. were defined beforehand. When a sketch was input to the method, each feature was detected and extracted repeatedly from that, and a solid model was obtained by combining 3D features. However, after a feature was extracted from a sketch, some restoration process was needed because of the next feature extraction. Since the restoration process was ad hoc, its programming tended to be difficult. In this paper, a new method is proposed. This method would formalize and simplify the restoration process by applying our inductive learning technique. Although the learning technique had an issue that properties of geometric elements would increase exponentially, this method could prevent the increase by applying our IFOG (Inductive Functional prOgramming for Geometric processing) technique. The effectiveness of this method is indicated by three examples in this paper.

KEYWORDS
Sketch; reconstruction; solid model; inductive learning; IFOG

1. Introduction

Generally sketches as line drawings are important tools for designers when they create new mechanical parts and so forth. A human can draw a sketch of a 3D object, and the other human would recognize the object from the sketch. This human behavior fascinated a lot of researchers who long for realizing artificial intelligence systems. If a system that can automatically reconstruct a 3D object as a solid model from a sketch is realized, it would be useful for designers and also some images might be automatically converted to 3D models. Recent years, the systems of deep learning in neural networks have been developed remarkably, and they have abilities to automatically recognize many things from images, e.g. [17]. If our proposed method in this paper cooperates with them, computers might not only recognize objects but also reconstruct 3D models from images.

Since decades, the systems to automatically reconstruct solid models from their sketches have been researched, and many methods have been proposed. However, these methods could not reconstruct solid models including curved faces. Recent years we proposed a method for the reconstruction, e.g. [20]. Basically humans can easily recognize sketches of primitives such as cuboids and cylinders. Our basic idea of the method was to divide a complex sketch into simple sketches. We called each simple sketch a feature of sketches. In the method, when a sketch is input, each feature was detected and extracted repeatedly from that, and a solid model was obtained by combining 3D features. However, it was a serious issue in the method that some restoration process was needed after some feature was extracted from a sketch because of the next feature extraction. Since the restoration process was ad hoc, its programming tended to be difficult. In this paper, a new method is proposed. This method would formalize and simplify the restoration process by applying our inductive learning technique. Although the learning technique had an issue that properties of geometric elements such as straight line segments and arcs would increase exponentially, this method could prevent the increase by applying our IFOG (Inductive Functional prOgramming for Geometric processing) technique, e.g. [21].

2. Related works

There are two kinds of related works concerned with this method proposed in this paper. One is for the reconstruction technique of solid models from sketches. The other is for the learning technique of the restoration process.

CONTACT Masaji Tanaka ✉ tanaka@mech.ous.ac.jp

For the reconstruction, the classification of techniques was well indicated in [3]. Originally line labelling technique was developed as Clowes-Huffman labelling, e.g. [2],[11]. In this labelling, sketches were projections of opaque trihedral polyhedrons placed in general viewpoints. Also, each line (segment) was labelled "+" or "-" or arrow. In a 3D object, "+" line can form a convex shape. "-" line can form a concave shape. An arrow line can form an occluding edge. From the labelling, four types of points were defined as L, W, Y and T junctions in sketches. For example, a Y junction consisting of three "+" lines can form a convex shape in a 3D object. Malik [13] interpreted sketches of curved objects in line labelling. He made the junction catalog of curved lines. Also, Cooper [7] presented more rich labelling scheme for curved objects by extending [13]. However, Malik and Cooper did not mention how to create solid models of curved objects from sketches including curved lines. In general, to infer hidden shapes that are not drawn in sketches of 3D objects is difficult. For example, Grimstead and Martin [10] proposed a method to reconstruct solid models from sketches of opaque trihedral polyhedrons by using skewed symmetry and so forth. Varley et al. tried to make depth reasoning of frontal geometry as sketches, e.g. [22–23]. Cao et al. [1] inferred hidden shapes by their unique method in polyhedrons. At the aspect of feature extraction for the reconstruction, Wang attempted to extract cylinders from sketches, e.g. [24]. In recent years, Company et al. proposed methods for detecting rounds, fillets [4] and ribs [5] from sketches. Plumed et al. proposed a method for detecting pockets and steps from sketches, e.g. [15]. Also, the fitness of elliptical arcs in sketches was discussed, e.g. [6]. Although the above mentioned methods could apply to partial reconstruction of sketches by using feature extraction, our first method for the reconstruction [20] and proposed method in this paper could handle more comprehensive sketches including curves.

For the learning technique, basically our technique would be a kind of machine learning, especially inductive learning, e.g. [14]. In general, after plural examples each of which consists of a problem and its solution are taught to a system of inductive learning, they would be generalized in the system. Then when a new problem is input to the system, its solution would be output automatically by applying the generalization. Iwama et al. proposed an inductive learning technique for learning mathematical sentences, e.g. [9],[12]. The technique could develop a program that acquires procedure to check the truth or the falsity of expressions and sentences such as "7 is a prime number". For developing the technique, he created internal variables when plural examples were generalized. The creation of them was effective for introducing learning technique to CAD applications because CAD programs were based on mathematics especially in geometry. Therefore, we applied his technique to our research. Our first application of the technique was a method to restore partial omissions in 2D mechanical drawings drawn in 2D CAD systems, e.g. [18–19]. In the method, each geometric element such as a straight line and each relationship of plural geometric elements were defined as a set of properties respectively. The restoration process of a partial omission of a 2D drawing could be generalized by using the changes of these properties. However, when the method was extended to more various kinds of partial omissions, the number of properties would increase exponentially. This exponential increase of properties was a serious issue for the method. In this paper, a new method is proposed. This method would solve the issue by using our IFOG (Inductive Functional prOgramming for Geometric processing) technique [21]. As the result, this method would handle more various kinds of sketches including curves than our previous method [20]. Here, IFOG is explained simply as follows. Geometric elements such as lines and the relationships of them are defined as classes in IFOG. Therefore, a sketch consists of their instances. Each class consists of properties. For example, the class of points can consist of two properties as point number and x-y coordinates. Since classes and instances are stored as text files, it is easy to update them in IFOG. When a user wants to add a new property to a class, he/she can write it in the text file of the class, and then the property can automatically reflect to all instances of the class. Also, he/she can make a new class as a text file. Therefore, whenever some new properties are needed in a class, they can be added simply to the class. As the result, the number of properties would be constantly minimized in IFOG.

In recent years, a great many machine learning systems have been proposed and developed. Also, for developing advanced CAD systems, many researches have applied machine learning. For example, Qin et al. [16] applied deep learning to classify 3D models of mechanical parts such as gears and screws for simpler design of them from internet. Cui et al [8] proposed a method to optimize dimensions of parts of ships by using Genetic Algorithm (GA). Especially, deep learning has become popular and has its applications in pattern matching, e.g. [17]. 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 puts the steps into a sequence of the steps. The learning technique of our proposed method in this paper can construct a procedure from example sequences of steps. For example, this method can learn a procedure to calculate x-y coordinates of the intersection of two straight lines by generalizing a few examples. Moreover, the procedure can be applied as a function to the other procedures, e.g. a procedure that can divide two lines into four lines at their intersection.

3. Whole algorithm to reconstruct solid models from sketches

The whole algorithm of our proposed method in this paper to reconstruct solid models from sketches including curves is illustrated in Fig. 1. In this method, a sketch is drawn in a 2D CAD and consists of straight lines, elliptical arcs and ellipses. The algorithm is explained with Example 1 illustrated in Fig. 2(a) as follows. In this example, a sketch of a cuboid and a sketch of a cylinder are handled as two features respectively. The definitions of the features and the sketch are indicated in detail in [20]. Before Example 1 is input to this method, the following three classes are defined.

Class Point: 1) Number 2) x-y coordinates;

Class Line: 1) Number 2) Kind 3) Two terminals 4) Major radius 5) Minor radius 6) Center point;

Class Relationship between two lines: 1) Number 2) Two lines 3) Connection point;

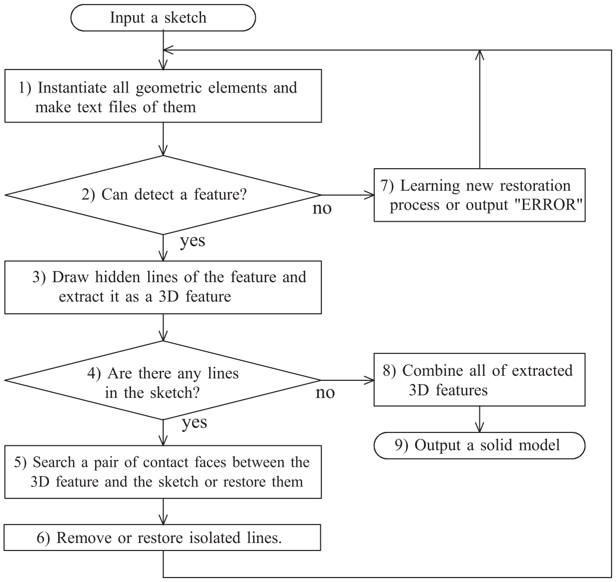

Input a sketch
1) Instantiate all geometric elements and make text files of them
2) Can detect a feature?
no
yes
7) Learning new restoration process or output "ERROR"
3) Draw hidden lines of the feature and extract it as a 3D feature
4) Are there any lines in the sketch?
no
yes
8) Combine all of extracted 3D features
9) Output a solid model
5) Search a pair of contact faces between the 3D feature and the sketch or restore them
6) Remove or restore isolated lines.

Figure 1. Whole algorithm to reconstruct solid models from sketches.

(1) When a sketch is input to this method, firstly all geometric elements are instantiated to the three classes and the text files of them are stored in a PC. Then each region that is a closed loop of lines is recognized. After Step 7, if some class is changed, its instances are also changed automatically by using IFOG [21].

(2) A sketch of a cylinder can be detected as a feature as in Fig. 2(b). If there are no features, Step 7 is executed.

(3) The hidden line of the feature ($f1$) can be drawn as a dotted arc as in Fig. 2(b). In Fig. 2(c), $f1$ is extracted and converted to a 3D cylinder.

(4) Features are extracted from the sketch repeatedly until there are no lines in it.

(5) Since it is found $c1$ of $f1$ is the contact face to the other feature, $c1$ can also be drawn as a dotted ellipse in the remained sketch as in Fig. 2(c).

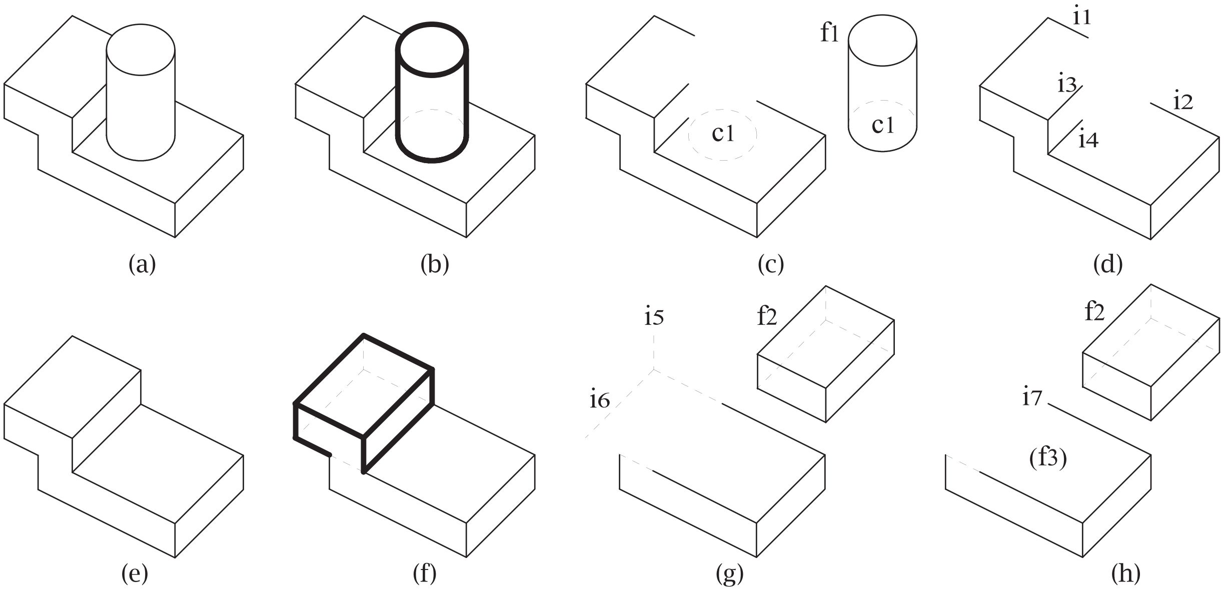


Figure 2. Example 1: (a) Example 1, (b) Detection of a cylinder, (c) Extraction of the cylinder, (d) Four isolated lines, (e) Restored sketch, (f) Detection of a cuboid, (g) Extraction of the cuboid, and (h) Remained lines to form a cuboid ($f3$).

(6) Each isolated line does not form any regions and features in this method. In Fig. 2(c), there are four isolated straight lines and one isolated dotted ellipse ($c1$). $c1$ can be removed as in Fig. 2(d). In this figure, four isolated lines are numbered ($i1$, $i2$, $i3$, $i4$). They have to be restored into normal lines because of the other feature extractions.

(7) In this step, our learning technique is applied. Its detailed explanation is described in the next chapter. From Fig. 2(d), the following processes are executed. Suppose that the four isolated lines can be restored as in Fig. 2(e). In Fig. 2(f), a sketch of a cuboid can be detected as a feature, and its hidden lines can be drawn. In Fig. 2(g), the feature ($f2$) is extracted as a 3D cuboid. This extraction is executed by removing all solid lines of $f2$ in Fig. 2(f). Therefore, there are four dotted lines including two isolated lines ($i5$, $i6$) in Fig. 2(g). In this method, an isolated dotted line that cannot connect to the other lines when it is extended is removed. Therefore, $i5$ and $i6$ are removed and then an isolated dotted line that connects to them is also removed. Consequently an isolated solid line ($i7$) appears as in Fig. 2(h). In this figure, it is necessary to restore a cuboid ($f3$) for making the solid model of Example 1 by applying the learning technique. If no features can be restored in this step, this method outputs "ERROR". In this case, there would be some mistake in input sketch.

(8) Suppose that three 3D features ($f1$, $f2$, $f3$) are obtained. They can be combined at their contact faces in accordance with Fig. 2(a). In detail, we apply the method of Varley et al. [22] to make a solid model from features. For example, a Y junction can be transformed into a cubic corner in this literature.

(9) The solid model of Example 1 could be output.

4. Algorithm of learning technique to restore features

Fig. 3 shows the algorithm of learning technique to restore features in our proposed method in this paper. In Example 1, both of two restorations in Fig. 2(d) and Fig. 2(h) would be summarized as the restoration of parallelograms. Therefore, the explanation of the algorithm is described by applying the learning of parallelograms from sketches as follows. Here, a problem is a sketch, and whether it is a parallelogram or not is the solution.

(1) Firstly, Fig. 4(a) drawn in a 2D CAD is input as Problem 1(Pb1) to this method. From Step 5, Fig. 4(b) is input as Pb2. In these figures, the numbers of points and lines are indicated. Fig. 4(c) and Fig. 4(d) illustrates the contents of text files of instances in Pb1 and Pb2 respectively. In these figures, for example, in point(Pb1).txt, the x-y coordinates of $P1$ is (25, 35). In line(Pb1).txt, since "Kind" is straight line constantly, this property is omitted, and three properties (4, 5, 6) concerned with curves are also omitted. For example, the two terminals of $L3$ are $P1$ and $P3$.

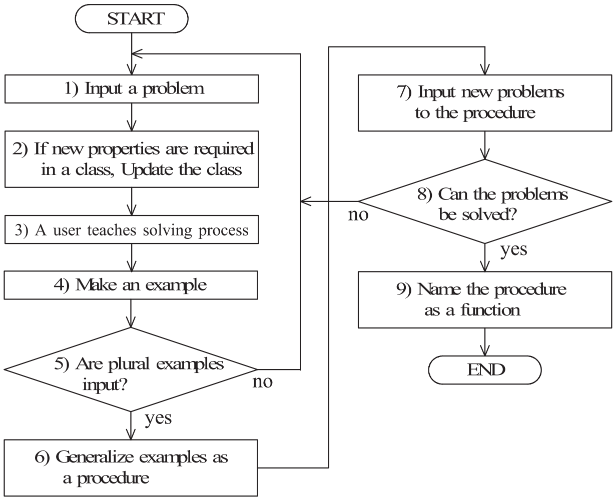

START
1) Input a problem
2) If new properties are required in a class, Update the class
3) A user teaches solving process
4) Make an example
5) Are plural examples input?
no
yes
6) Generalize examples as a procedure
7) Input new problems to the procedure
8) Can the problems be solved?
no
yes
9) Name the procedure as a function
END

Figure 3. The algorithm of learning technique to restore features.

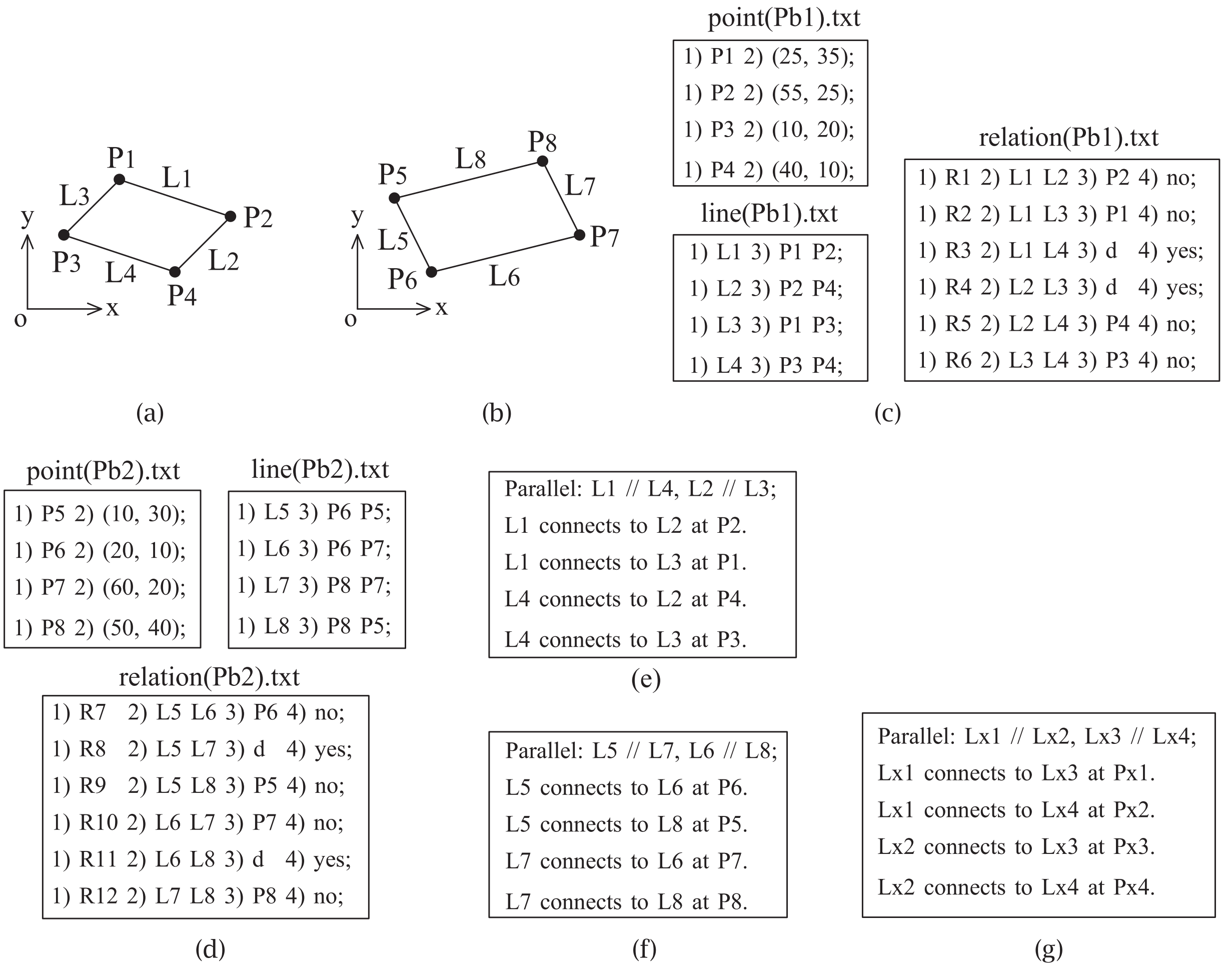
(2) In relation(Pb1).txt of Fig. 4(c), a new property is added as "4) Parallel" because of searching parallelograms. IFOG [21] enables this addition. Also, "d" means "default".

(3) A user can input relations of properties step by step to learn parallelograms. The relations can be summarized as in Fig. 4(e). For example, this teaching can be expressed as Q & A as follows. Q1) Show all parallel lines. A1) L1 // L4, L2 // L3. Q2) Show intersection between L1 and L2. A3) P2. . . . These questions are already set in this method, and these answers can be obtained from Fig. 4(c).

(4) From Fig. 4(e), it is found that Pb1 is a parallelogram. As the result, one example can be made from Pb1. In the same way, the other example can be made from Pb2 by applying Fig. 4(f).

(5) After those two examples are input, Step 6 is executed.

(6) The properties between Fig. 4(e) and Fig. 4(f) can be generalized as Fig. 4(g) as follows. At the first sentences of them, "L1 // L4, L2 // L3" and "L5 // L7, L6 // L8", since $L1$ and $L5$ are different lines, these line numbers can become a variable as $Lx1$. Therefore, the sentences can be generalized as "Lx1 // Lx2, Lx3 // Lx4". These four variable lines are reflected to the other sentences as in this figure. In the same way, there are four variable points (Px1, Px2, Px3, Px4) in the figure. As the result, two examples can



point(Pb1).txt

```
1) P1 2) (25, 35);
1) P2 2) (55, 25);
1) P3 2) (10, 20);
1) P4 2) (40, 10);
```

line(Pb1).txt

```
1) L1 3) P1 P2;
1) L2 3) P2 P4;
1) L3 3) P1 P3;
1) L4 3) P3 P4;
```

relation(Pb1).txt

```
1) R1 2) L1 L2 3) P2 4) no;
1) R2 2) L1 L3 3) P1 4) no;
1) R3 2) L1 L4 3) d  4) yes;
1) R4 2) L2 L3 3) d  4) yes;
1) R5 2) L2 L4 3) P4 4) no;
1) R6 2) L3 L4 3) P3 4) no;
```

point(Pb2).txt

```
1) P5 2) (10, 30);
1) P6 2) (20, 10);
1) P7 2) (60, 20);
1) P8 2) (50, 40);
```

line(Pb2).txt

```
1) L5 3) P6 P5;
1) L6 3) P6 P7;
1) L7 3) P8 P7;
1) L8 3) P8 P5;
```

relation(Pb2).txt

```
1) R7  2) L5 L6 3) P6 4) no;
1) R8  2) L5 L7 3) d  4) yes;
1) R9  2) L5 L8 3) P5 4) no;
1) R10 2) L6 L7 3) P7 4) no;
1) R11 2) L6 L8 3) d  4) yes;
1) R12 2) L7 L8 3) P8 4) no;
```

```
Parallel: L1 // L4, L2 // L3;
L1 connects to L2 at P2.
L1 connects to L3 at P1.
L4 connects to L2 at P4.
L4 connects to L3 at P3.
```

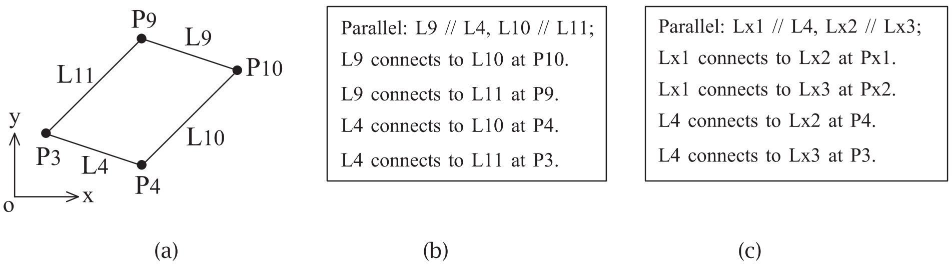
```
Parallel: L5 // L7, L6 // L8;
L5 connects to L6 at P6.
L5 connects to L8 at P5.
L7 connects to L6 at P7.
L7 connects to L8 at P8.
```

```
Parallel: Lx1 // Lx2, Lx3 // Lx4;
Lx1 connects to Lx3 at Px1.
Lx1 connects to Lx4 at Px2.
Lx2 connects to Lx3 at Px3.
Lx2 connects to Lx4 at Px4.
```

Figure 4. Two problems of parallelograms: (a) Problem 1(Pb1), (b) Problem 2(Pb2), (c) Three text files expressing properties in Pb1, (d) Three text files expressing properties in Pb2, (e) Solution of Pb1, (f) Solution of Pb2, and (g) Generalized solution from (e) and (f).

be generalized inductively as a procedure to recognize a parallelogram from four lines. In this case, only two examples are required for the generalization because Pb1 and Pb2 are quite different. If similar examples are learned, more examples are required as follows. Fig. 5(a) shows Problem 3(Pb3). In Pb3, only the lengths of L2 and L3 are doubled from Pb1. Therefore, *P*3, *P*4, *L*4 are the same between Pb1 and Pb3. Fig. 5(b) shows the solution of Pb3. Here, L9 // L4 is equals to L4 // L9 when they are processed in this method. When Fig. 4(e) and Fig. 5(b) are generalized, Fig. 5(c) is obtained. However, since *P*3, *P*4, *L*4 are fixed in this figure, this generalization cannot be completed. When some new example is added and all of these three numbers of geometric elements are changed into variables, this generalization can be completed.

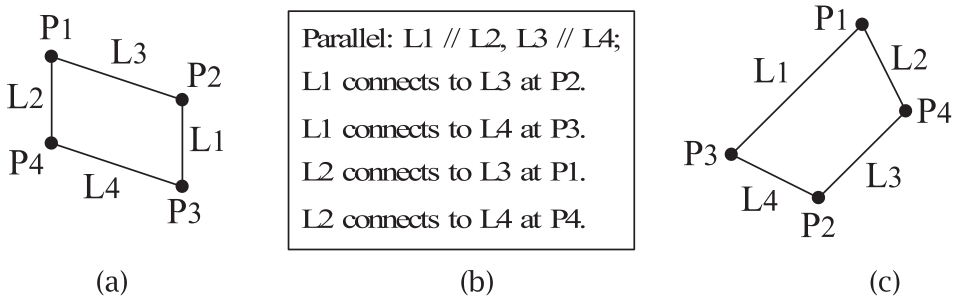
(7) Fig. 6(a) shows Problem 4 (Pb4). When Pb4 is input to this method, all of variable lines and variable points in Fig. 4(g) can be replaced into real lines and



```
Parallel: L9 // L4, L10 // L11;
L9 connects to L10 at P10.
L9 connects to L11 at P9.
L4 connects to L10 at P4.
L4 connects to L11 at P3.
```

```
Parallel: Lx1 // L4, Lx2 // Lx3;
Lx1 connects to Lx2 at Px1.
Lx1 connects to Lx3 at Px2.
L4 connects to Lx2 at P4.
L4 connects to Lx3 at P3.
```

Figure 5. Problem 3(Pb3): (a) Pb3, (b) Solution of Pb3, and (c) Generalized solution from (b) and Fig. 4(e).



Parallel: L1 // L2, L3 // L4;
L1 connects to L3 at P2.
L1 connects to L4 at P3.
L2 connects to L3 at P1.
L2 connects to L4 at P4.

Figure 6. New two problems of parallelograms: (a) Problem 4, (b) Solution of Problem 4, and (c) Problem 5.

points of Pb4 as in Fig. 6(b). Therefore, it is found that Pb4 is a parallelogram. Fig. 6(c) illustrates Problem 5 (Pb5). When Pb5 is input to this method, it is found that Pb5 is not a parallelogram because there are not two pairs of parallel lines.

(8) Since Pb4 and Pb5 can be solved correctly, “yes” is selected.

(9) The generalized procedure can become a function of computer programs, and it can be named “Parallelogram_Recognition(Lx1, Lx2, Lx3, Lx4)”.

5. Restoration of parallelograms

To restore Fig. 2(d) and Fig. 2(h), three restoration types of parallelograms are required. These three types can be modeled as in Fig. 7. In Type 1, it is found that the lengths of two isolated lines are too short to form a parallelogram. In Type 2, it is found that one line is lacked between two terminals of two isolated lines. In Type 3, it is found that these two types are both included. Therefore, in this chapter, the learning of restoration of Type 3 is explained in detail as follows. To apply our learning technique to Type 3, “Length” has to be added as a new property to class line as follows.

Class Line: 1) Number 2) Kind 3) Two terminals 4) Major radius 5) Minor radius 6) Center point 7) Length;

Also, “Contact lines” had better be added as a new property to class point as follows.

Class Point: 1) Number 2) x-y coordinates 3) Contact lines;

Fig. 8(a) illustrates Restoration Problem 1 (RP1) that corresponds to Type 3. All initial instances of that are illustrated in Fig. 8(b). In line(RP1).txt of this figure, the properties of “2), 4), 5), 6)” are omitted because RP1 does not include any curves. Although there may be many ways for this restoration, a simple way is applied here. In this method, a user can do sequential Q & A as follows.

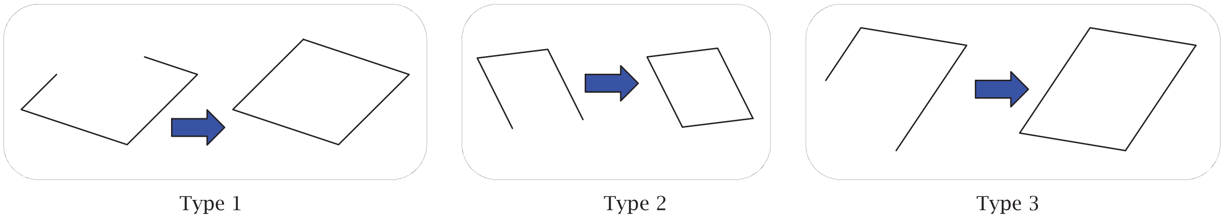
Q1) Show all parallel lines. A1) L2 L3 Q2) Which is longer? A2) L3 Q3) Show isolated point of L3. A3) P4 Q4) Copy L1 from P4 as L4. A4) (indicated on monitor) Q5) Show isolated point of L4. A5) P5 Q6) Extend L2 until P5. A6) (monitor).

These questions are already set in this method, and these answers can be obtained from Fig. 8(b). When A4 or A6 is executed, the contents of Fig. 8(b) are updated step by step. As the result, a parallelogram is completed as in Fig. 8(c). The confirmation of that can be executed by using Parallelogram_Recognition(L1, L2, L3, L4). Fig. 9(a) illustrates Restoration Problem 2 (RP2) of Type 3. The user can do the same sequential Q & A as follows.

Q1) Show all parallel lines. A1) L5 L6 Q2) Which is longer? A2) L6 Q3) Show isolated point of L6. A3) P6 Q4) Copy L7 from P6 as L8. A4) (indicated on monitor) Q5) Show isolated point of L8. A5) P10 Q6) Extend L5 until P10. A6) (monitor).

As the result, a parallelogram is completed as in Fig. 9(b). These two Q & A operations can be generalized inductively as follows.

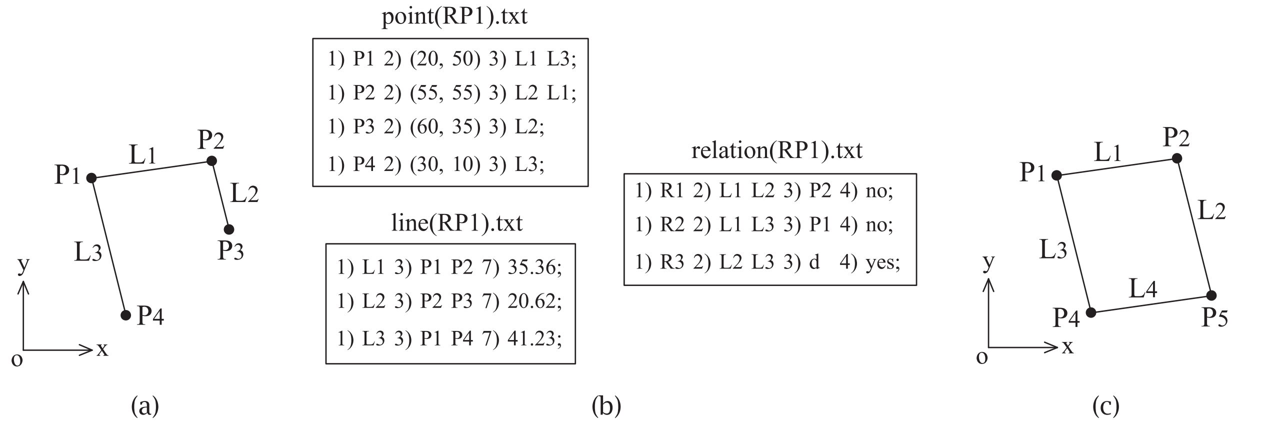
Q1) Show all parallel lines. A1) Lx1 Lx2 Q2) Which is longer? A2) Lx1 Q3) Show isolated point of Lx1. A3) Px1 Q4) Copy Lx3 from Px1 as Lx4. A4) (indicated on monitor) Q5) Show isolated point of Lx4. A5) Px2 Q6) Extend Lx2 until Px2. A6) (monitor).


Type 1

Type 2

Type 3

Figure 7. Three restoration types of parallelograms.



point(RP1).txt

```
1) P1 2) (20, 50) 3) L1 L3;
1) P2 2) (55, 55) 3) L2 L1;
1) P3 2) (60, 35) 3) L2;
1) P4 2) (30, 10) 3) L3;
```

line(RP1).txt

```
1) L1 3) P1 P2 7) 35.36;
1) L2 3) P2 P3 7) 20.62;
1) L3 3) P1 P4 7) 41.23;
```

relation(RP1).txt

```
1) R1 2) L1 L2 3) P2 4) no;
1) R2 2) L1 L3 3) P1 4) no;
1) R3 2) L2 L3 3) d   4) yes;
```

Figure 8. Restoration Problem 1(RP1): (a) RP1, (b) Three text files expressing properties in RP1, and (c) Solution as a parallelogram.

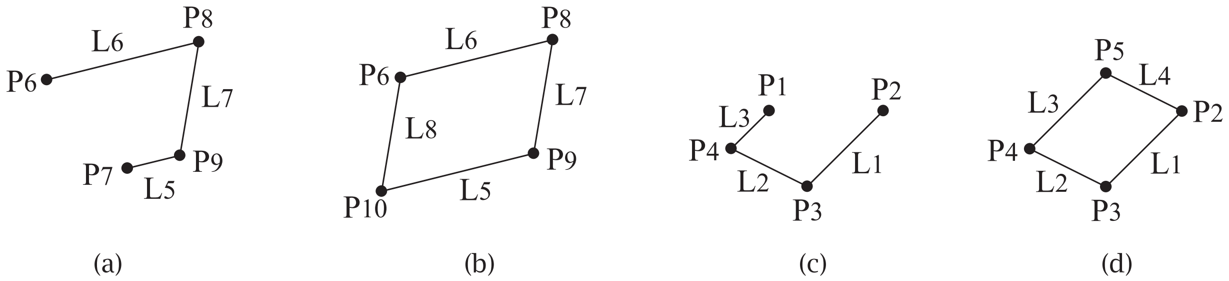


Figure 9. Restoration Problem 2(RP2) and a new problem: (a) RP2, (b) Solution of RP2, (c) A new restoration problem, and (d) Solution of the problem.

Fig. 9(c) illustrates a new problem. The generalized operations are applied to the problem as follows.

Q1) Show all parallel lines. A1) L1 L3 Q2) Which is longer? A2) L1 Q3) Show isolated point of L1. A3) P2 Q4) Copy L2 from P2 as L4. A4) (indicated on monitor) Q5) Show isolated point of L4. A5) P5 Q6) Extend L3 until P5. A6) (monitor).

As the result, a parallelogram is completed as in Fig. 9(d). In conclusion, it is found that Fig. 2(d) and Fig. 2(h) can be restored by applying the learning technique of this method.

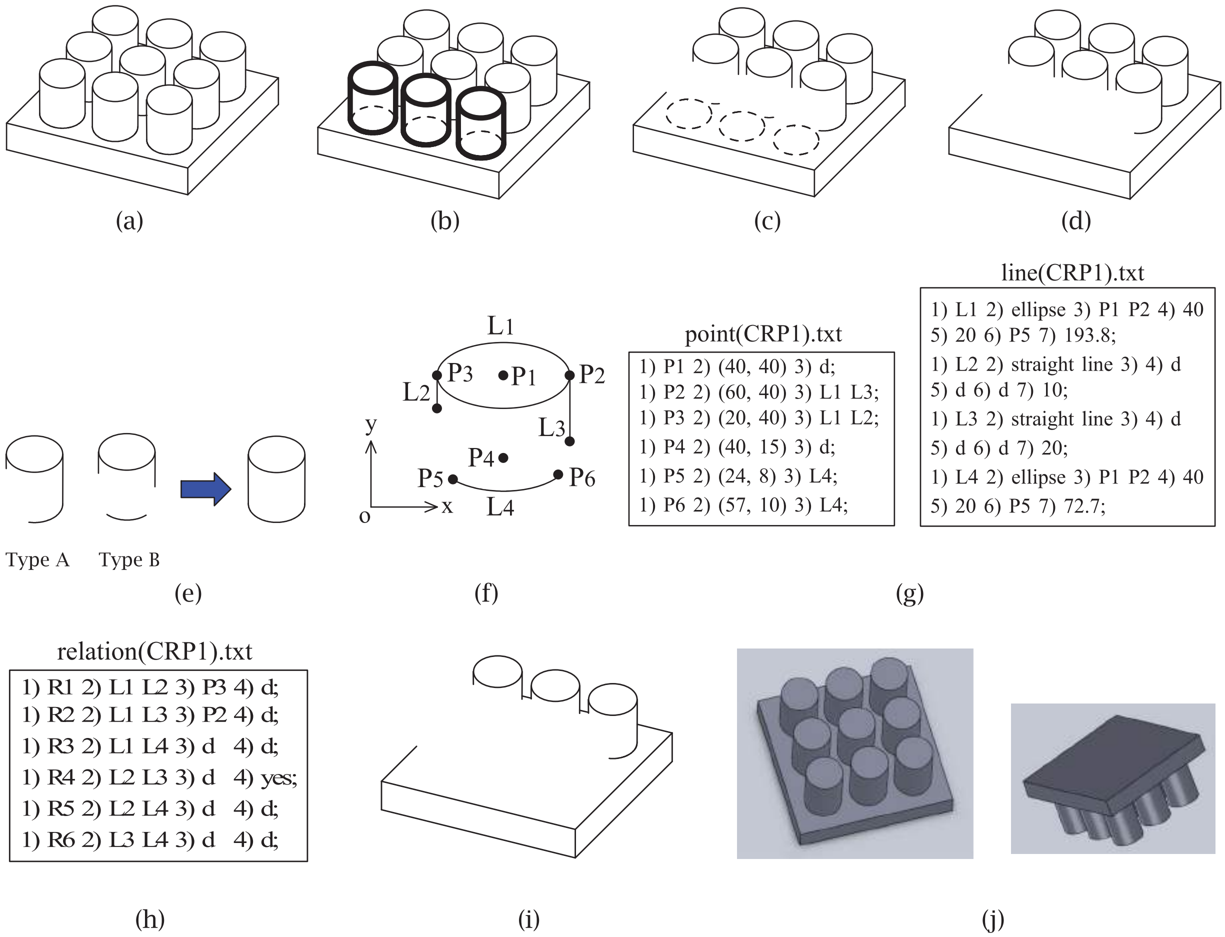
6. Examples

In this chapter, two examples are indicated. Fig. 10(a) illustrates Example 2 that imagines a bubble wrap. Firstly, it is obvious that three cylinders can be detected respectively as in Fig. 10(b). After they are extracted, Fig. 10(c) is obtained. In this figure, there are three isolated dotted ellipses as contact faces corresponding to the three cylinders. In Fig. 10(d), they are removed. There are two types of cylinders that need restoration processes as in Fig. 10(e). Since Type A is similar to Type 1 in Fig. 7, here, the restoration of Type B is explained as follows. Fig. 10(f) shows Curves Restoration Problem 1 (CRP1). Fig. 10(g) and Fig. 10(h) show all initial instances of CRP1. In these figures, ellipse ($L1$) is not divided by any lines because it is meaningless. The detailed explanation of that is indicated in [20]. For the restoration of CRP1, a user can do sequential Q & A as follows.

Q1) Show one ellipse. A1) L1 Q2) Show two straight lines each of which contacts to L1. A2) L2 L3 Q3) Is L2 parallel to L3? A3) yes Q4) Is there one elliptical arc below L2 and L3? A4) yes Q5) Show the arc. A5) L4 Q6) Is L4 a bottom part of L1? A6) yes Q7) Extend L4 until it becomes a half ellipse. A7) (indicated on monitor). Q8) Extend L2 and L3 until each of them contacts tangentially to L4. A8) (monitor).

When the other example is input, the restoration process of Type B can be generalized. As the result, in Fig. 10(d), three cylinders can be restored and extracted as in Fig. 10(i). In this figure, there are two broken cylinders that cannot be applied to the restoration process described above because of lacking arcs. In this case, since a user knows that all heights of cylinders are the same in Example 2, he/she can teach the height as in the following generalized process.

Q1) Show one ellipse. A1) Lx1 Q2) Show two straight lines each of which contacts to Lx1. A2) Lx2 Lx3 Q3) Is Lx2 parallel to Lx3? A3) yes Q4) Is there one elliptical



point(CRP1).txt

```
1) P1 2) (40, 40) 3) d;
1) P2 2) (60, 40) 3) L1 L3;
1) P3 2) (20, 40) 3) L1 L2;
1) P4 2) (40, 15) 3) d;
1) P5 2) (24, 8) 3) L4;
1) P6 2) (57, 10) 3) L4;
```

line(CRP1).txt

```
1) L1 2) ellipse 3) P1 P2 4) 40
5) 20 6) P5 7) 193.8;
1) L2 2) straight line 3) 4) d
5) d 6) d 7) 10;
1) L3 2) straight line 3) 4) d
5) d 6) d 7) 20;
1) L4 2) ellipse 3) P1 P2 4) 40
5) 20 6) P5 7) 72.7;
```

relation(CRP1).txt

```
1) R1 2) L1 L2 3) P3 4) d;
1) R2 2) L1 L3 3) P2 4) d;
1) R3 2) L1 L4 3) d  4) d;
1) R4 2) L2 L3 3) d  4) yes;
1) R5 2) L2 L4 3) d  4) d;
1) R6 2) L3 L4 3) d  4) d;
```

Figure 10. Example 2: (a) Example 2, (b) Detection of three cylinders, (c) Extraction of the cylinders, (d) Removal of three dotted ellipses from (c), (e) Two restoration types of a cylinder, (f) Curves Restoration Problem 1(CRP1) (g) Point and line files of CRP1, (h) Relation file of CRP1, (i) Extraction of the three cylinders of (d), and (j) Two overviews of the solid model of Example 2.

arc below Lx2 and Lx3? A4) no Q5) Show the height of each cylinder. A5) 15 Q6) Extend Lx2 and Lx3 until their lengths become 15. A6) (indicated on monitor). Q7) Add one half ellipse which is a bottom part of Lx1 from the terminal of Lx2 to the terminal of Lx3 tangentially. A7) (monitor).

From Q5, this generalized process diverges into two processes by the value of A4. Consequently, nine cylinders can be detected and extracted from Example 2, and then a cuboid can be detected. When these extracted 3D features are combined, the solid model of Example 2 can be obtained as in Fig. 10(j).

This method can be applied to not only mechanical parts but also more kinds of objects. Fig. 11(a) illustrates Example 3 that imagines a bunch of grapes on a table. In this example, since an ellipse corresponds to a grain of grapes, firstly a user has to teach that each ellipse can be a feature of ellipsoid when a sketch almost consists of many ellipses and elliptical arcs. A restoration process of the grapes can be described as follows.

Q1) Is there a mass of elliptical arcs and ellipses? A1) yes Q2) Are there any ellipses? A2) yes Q3) Show one. A3) Lx1 Q4) Extract Lx1 as an ellipsoid A4) (indicated on monitor) Go to Q2 until the answer of A2 is "no". Q5) Are there any elliptical arcs each of which can form an ellipse by its extension without intersecting the other lines? A5) yes Q6) Show one. A6) Lx2 Q7) Make an ellipse by extending Lx2. A7) (monitor) Q8) Extract Lx2 as an ellipsoid A8) (monitor) Go to Q5 until the answer of A5 is "no".

In this generalized process, there are two loops of subprocesses. This method can generalize these loops such as conventional program languages. From Fig. 11(b) to Fig. 11(j), the processes of extracting ellipses as ellipsoids sequentially by using the generalized restoration process are illustrated. After all ellipses are extracted, a cuboid can be extracted as in Fig. 11(k). In the solid model of Example 3, many ellipsoids are contacted to each other in accordance with Fig. 11(a). Fig. 11(l) shows an overview of the solid model of Example 3.

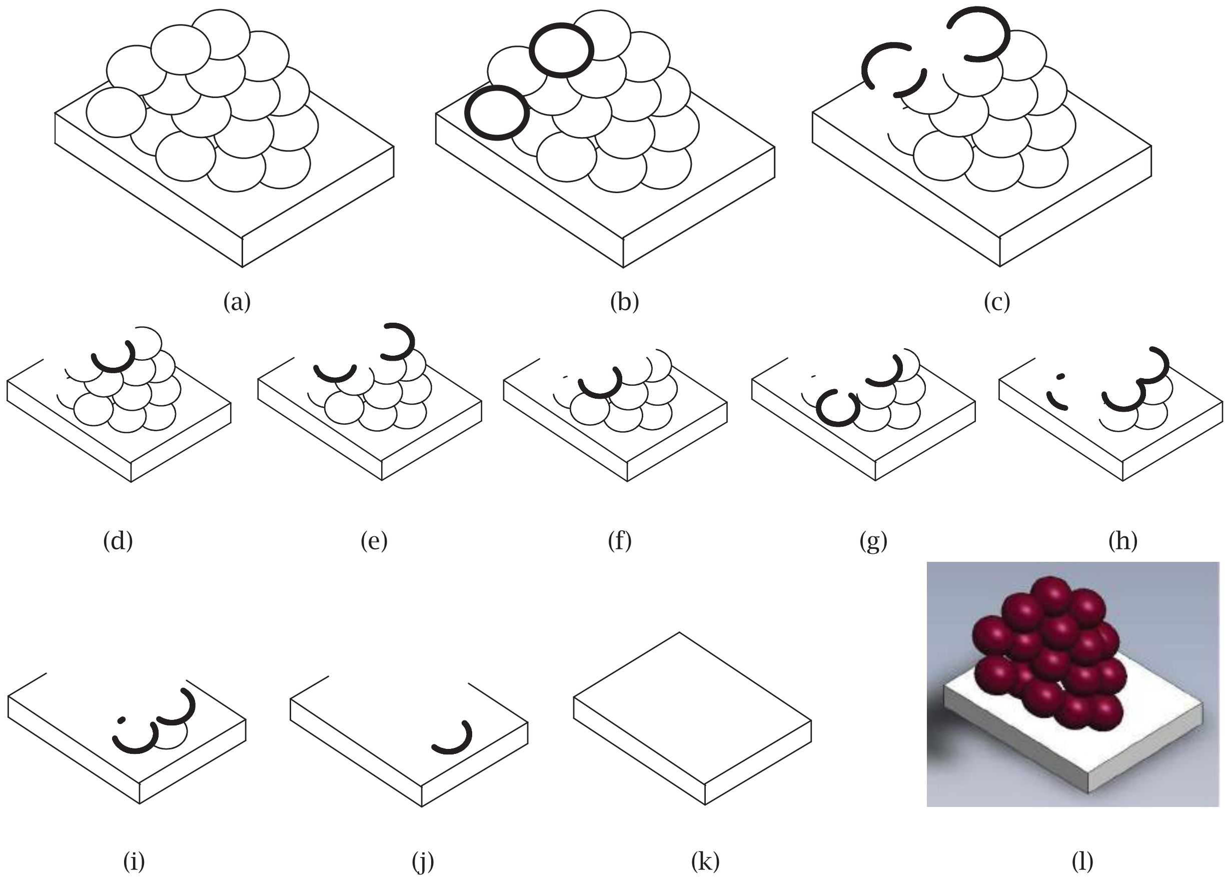


Figure 11. Example 3: (a) Example 3, (b) Detection of two ellipses (c) Their extraction and detection of three elliptical arcs, (d) Their extraction and detection of one elliptical arc, (e) Its extraction and detection of two elliptical arcs, (f) Their extraction and detection of one elliptical arc, (g) Its extraction and detection of two elliptical arcs, (h) Their extraction and detection of four elliptical arcs, (i) Their extraction and detection of three elliptical arcs, (j) Their extraction and detection of one elliptical arc, (k) Its extraction and detection of one cuboid, and (l) An overview of solid model.

7. Discussion

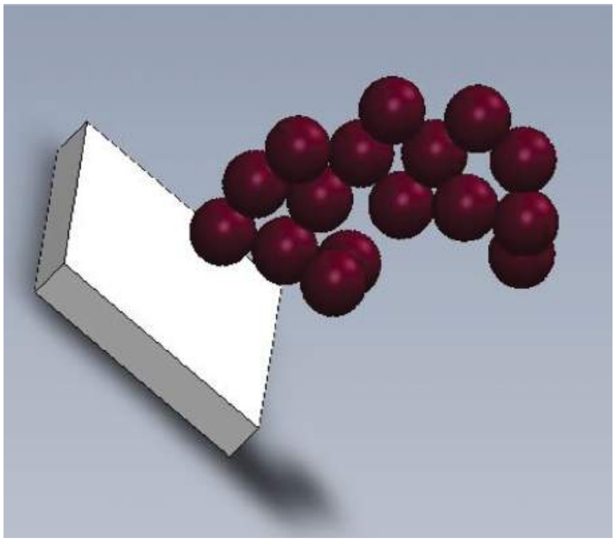


Figure 12. The other overview of the solid model in Fig. 11(c).

Since learning process is included in Step 7 of Fig. 1, the reconstruction of solid models from sketches would be semi-automatically executed in this method. If a lot of restoration processes have been learned, more sketches would be automatically converted to solid models. The learning technique can learn geometric shapes such as parallelograms. Also, that can learn a procedure consisting of sequential steps as operations in CAD such as restoring parallelograms, the sketches of cylinders and the sketches of ellipsoids. However, there are still many issues in this method. First, in the learning system, how to set questions and how to understand them by a computer would be an issue. Second, how to extract appropriate three or four lines from a sketch when a parallelogram has to be restored would be an issue. Third, although Example 3 could be converted as a solid model as in Fig. 11(l), from the other viewpoint, the solid model does not form a bunch of grape such as in Fig. 12. Therefore, more extensive learning technique would be required for making more adequate solid models from Example 3. The realization of such learning is an issue. However, since the learning technique of this method would have logical flexibility such as mathematics, that would have enough potential to solve those issues.

8. Conclusions

In this paper, a new method is proposed for the reconstruction of solid models from sketches including curves. Our inductive learning technique and IFOG are applied to this method for the restorations of broken sketches which usually occur after each feature is extracted. Since the broken sketches are often restored by restoring parallelograms, the algorithm of the learning technique is explained by applying the restoration of that. Finally, two remarkable examples and several issues of this method could be indicated.

ORCID

Masaji Tanaka http://orcid.org/0000-0002-5266-9182
Yuki Takamiya http://orcid.org/0000-0002-8950-8833
Naoki Tsubota http://orcid.org/0000-0003-4826-5262
Shigeyuki Asanuma http://orcid.org/0000-0002-3799-8033

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.; Varley, C.; Plumed, R.; Martin, R.: Perceiving Ribs in Single-View Wireframe Sketches of Polyhedral Shapes, G. Bebis et al. (Eds.): ISVC 2012, Part II, LNCS 7432, 2012, 557–567, http://dx.doi.org/10.1007/978-3-642-33191-6_55

[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] Cooper, M.C.: A rich discrete labeling scheme for line drawings of curved objects, IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(4), 2008, 741–745, http://dx.doi.org/10.1109/TPAMI.2007.70835.

[8] 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

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

[10] 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.

[11] Huffman, D. A.: Impossible Objects as Nonsense Sentences, Machine Intelligence 6, New York American Elsevier, 1971, 295–323.

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

[13] Malik, J.: Interpreting Line Drawings of Curved Objects, International Journal of Computer Vision, 1, 1987, 73–103, http://dx.doi.org/10.1007/BF00128527

[14] 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

[15] 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

[16] 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.

[17] 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

[18] 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

[19] 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

[20] 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

[21] Tanaka, M.; Takamiya, Y.; Tsubota, N.; Iwama, K.: IFOG: Inductive Functional Programming for Geometric Processing, Computer-Aided Design & Applications, 13(3), 2016, 389–396. http://dx.doi.org/10.1080/16864360.2015.1114397

[22] 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.

[23] 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

[24] Wang, W.: A Regular Curved Object's CSG-Rep Reconstruction from a Single 2D Line Drawing, Proceedings of SPIE, 1608(1), 1991, 119–127.