Title:
Introduction of Learning Techniques for Creating Solid Models from Sketches Including Curves

Authors:
Masaji Tanaka, tanaka@mech.ous.ac.jp, Okayama University of Science
Yuki Takamiya, takamiya@mech.ous.ac.jp, Okayama University of Science
Naoki Tsubota, tsubota@mech.ous.ac.jp, Okayama University of Science
Shigeo Asanuma, asanuma@mech.ous.ac.jp, Okayama University of Science
Kenzo Iwama, whatisiwama@yahoo.co.jp, EngiCom Corporation

Keywords:
Inductive Learning, IFOG, Sketch, Solid Model

DOI: 10.14733/cadconfP.2016.200-204

Introduction:
Since several years, the authors have developed a method to automatically convert sketches as 2D line drawings including curved lines into 3D models [1]. However, it was difficult and time consuming to develop the experimental system of the method because various kinds of geometric pattern matching processes had to be programmed. On the other hand, since decades, machine learning techniques have been developing very quickly. Since the techniques are effective for pattern matching, we apply our inductive learning techniques, e.g. [2] to the method in our new method proposed in this paper. There are many machine learning systems. Especially, deep learning has become popular and has its applications in pattern matching, e.g. [3]. 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 method could construct a procedure from example sequences of steps. For example, our 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 at their intersection. Although huge amount of properties of geometric elements would be handled to use the learning techniques, our IFOG technique, e.g. [4] would prevent the exponential increase of the properties. As the result, our method can apply to more various kinds of sketches.

Main Idea:
In our method, sketches are perfectly drawn in 2D CAD systems. A 3D object is assumed to be viewed from a general viewpoint that reveals all edges and vertices in a sketch. The lines drawn in the sketches consist of ellipses, arcs and straight lines. Hidden lines of the 3D objects are not drawn in the sketches. A point is defined as an endpoint of each line. Between sketches and solid models, a point can correspond to a vertex, a line can correspond to an edge. When a 3D object drawn in a sketch is placed in Cartesian coordinate system, each of straight edges and the axes of curved edges are parallel to one of the axes of the system. Fig. 1 illustrates Example 1 that is a sketch of a 3D object. This sketch is drawn in x-y coordinate system. Fig. 2 illustrates all numbers of points and lines in Example 1. A line is not divided at T junction in this method. For example, since P_6 and P_{10} are T junctions, L_9 and L_{13} do not be divided into two lines respectively. In this paper, three sketches of features are defined as in Fig. 3. In this figure, a sketch of cuboid consists of three parallelograms. A sketch of cylinder consists of an ellipse, two parallel straight lines and an arc. A sketch of round hole consists of an ellipse and an arc. Fig. 4 illustrates the summary of the algorithm of the conversion of sketches into 3D models in our method. In Example 1, firstly a round hole can be recognized and extracted. Next, a

cylinder can be recognized and extracted as in Fig. 5. Since any features do not be extracted from this figure, a feature as a cuboid should be restored.

Since there are many patterns of broken cuboids such as in Fig. 6, the introduction of learning techniques would be effective for their automatic restorations. In the case of Fig. 5, a user can teach that L_{10} and L_{13} are extended until they are connected at a new point (P_{15}) as in Fig. 7. This learning can be generalized as the changes of properties of lines and points. In our method, the class of each geometric element is defined as a set of properties, and class files and their instances files are stored as text files in a PC. The class of point, line and the relationship of two lines are defined as follows.

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

Class Line: 1) Number 2) Two terminals;

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

For example, the properties of P_{10} are "1)10 2)(20,50)". The properties of L_{12} are "1)12 2)P1, P10". The properties of the relationships between L_{10} and L_{12} are "1)1 2)L10, L12 3)".

From Fig. 5 to Fig. 7, the properties of L_{10} and L_{12} and their relationships are changed as in Fig. 8. In the same way, another example can be learned as the changes of properties. After two or more examples are learned, they can be generalized such as in Fig. 9. In this figure, xn (n=1, … , 8) is a variable of numbers. After the generalization, when a new problem like Fig. 8 is input, the solution can be output automatically by changing variables into real numbers. Moreover, for the restoration of features, users can add some properties to classes interactively, and then the properties can be reflected to their instances automatically in our method by applying IFOG. This interactivity is derived from the flexibility of text files. Since 2D cuboids, 2D round holes and 2D cylinders are defined as three classes, three features are extracted step by step in Example 1. A 3D cuboid is made from a feature of 2D cuboid. Also, the 3D cuboid becomes the base part of the solid model of Example 1 because it can form a cubic corner at P_4, e.g. [5]. P_4 corresponds to V_4 in 3D space. In the same way, L_i corresponds to E_i, parallelograms correspond to 3D rectangles and ellipses correspond to 3D circles. Each of these 3D geometric elements is defined as a class in our method. So a 2D round hole and a 2D cylinder can convert easily to a 3D round hole and a 3D cylinder respectively, and they can place on the 3D cuboid exactly. In conclusion, the solid model of Example 1 can be obtained.

Fig. 10(a) illustrates Example 2 that imagines a bubble wrap. In our method, firstly three cylinders can be recognized as in Fig. 10(b) and extracted as in Fig. 10(c). In this figure, the right side cylinder can be restored as in Fig. 10(d) by extending an arc and a straight line such as the restoration of a cuboid in Example 1. After the cylinder is extracted, a cylinder behind it can be restored as in Fig. 10(e). In Fig. 10(f), the cylinder is extracted. In this figure, it is difficult to restore any cylinders. But if a user teaches that the height of all cylinders are the same, two cylinders can be restored as in Fig. 10(g). After these cylinders are extracted, two cylinders behind them can be restored and extracted as in Fig. 10(h). In this figure, two isolated lines can be deleted because they obviously do not form any features. So a cuboid can be recognized as in Fig. 10(i). As the result, the solid model of Example 2 can be obtained as in Fig. 10(j). In the future, we want to try rougher sketches such as in Fig. 11. This figure illustrates a sketch of a bunch of grapes on a table. If a user teaches that each ellipse corresponds to a sphere, the solid model of Fig. 10 might be obtained.

Conclusion:

In this paper, we propose a method to create solid models from sketches including curves. The effectiveness of the method would be summarized as follows.

(1) Since various kinds of restoration patterns of sketches of features could be learned, more humanized sketches such as Example 2 whose conversion would be difficult for conventional methods can be applied.

(2) By introducing IFOG technique, the handling of geometric elements could become simpler and clearer especially for the learning process of our method.

References:
[1] 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
[2] 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.
[3] 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
[4] Tanaka, M.; Takamiya, Y.; Tsubota, N.; Iwama, K: IFOG: Inductive Functional Programming for Geometric Processing, Computer-Aided Design & Applications, 13(3), 2015, 389-396. http://dx.doi.org/10.1080/16864360.2015.1114397
[5] Varley, P. A. C.; Martin, R. 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.

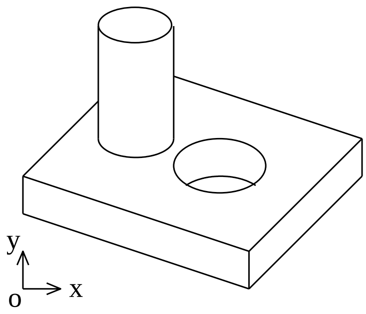


Fig. 1: Example 1.

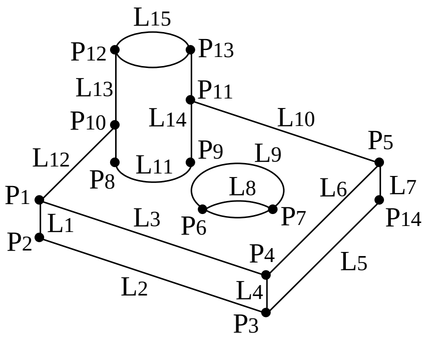


Fig. 2: All of points and lines in Example 1.

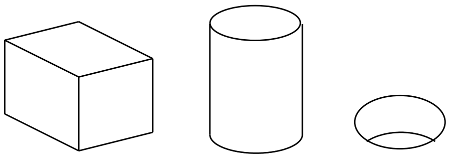


Fig. 3: Three sketches of features.

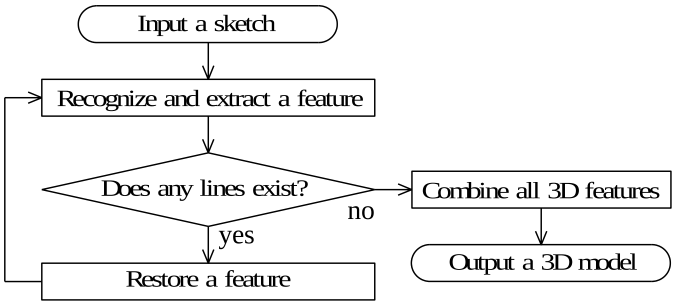


Fig. 4: Summarized algorithm of this method.

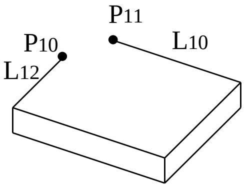


Fig. 5: Broken cuboid in Ex. 1.

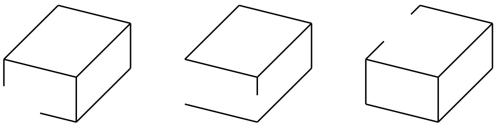


Fig. 6: Three patterns of broken cuboids.

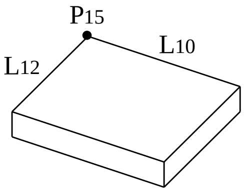


Fig. 7: Restored cuboid.

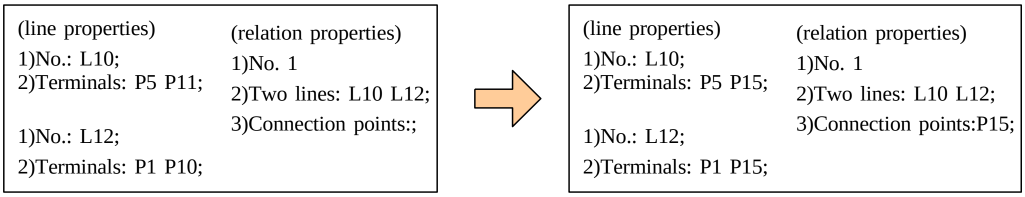


Fig. 8: Restoration process expressed by properties from Fig. 5 to Fig. 7.

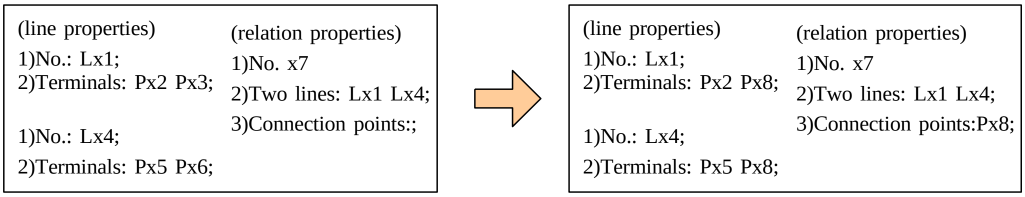
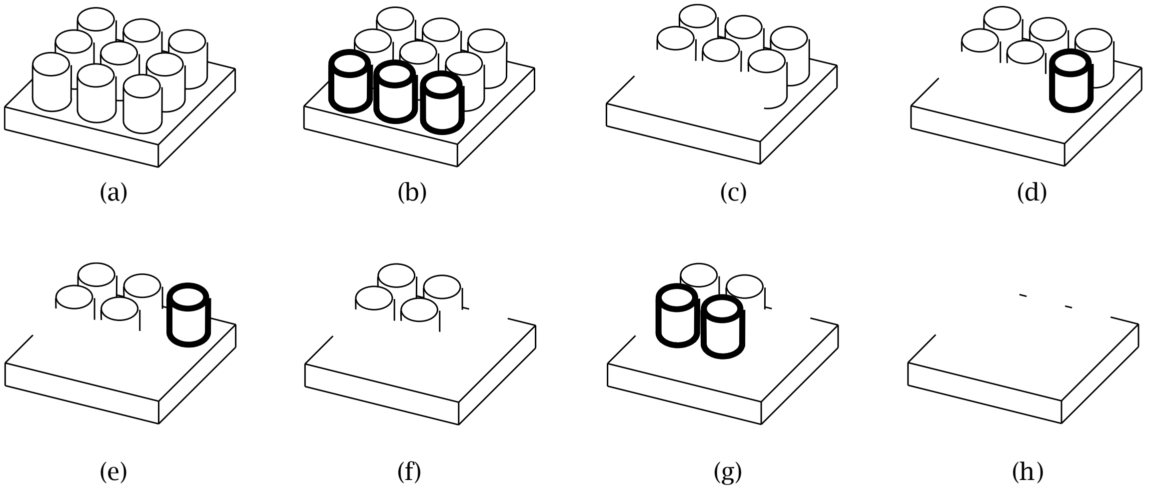
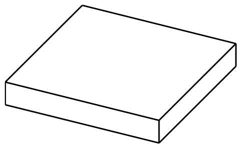
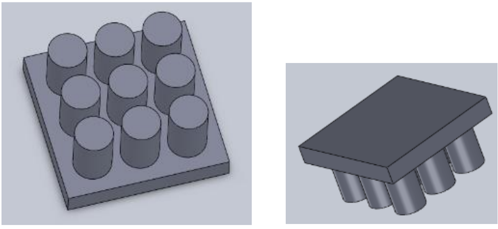


Fig. 9: Generalized restoration process from Fig. 8 and another example.





(i)



(j)

Fig. 10: Example 2: (a) Example 2, (b) Recognition of three cylinders, (c) Extraction of the cylinders, (d) Recognition of a cylinder, (e) Extraction of the cylinder and recognition of another cylinder, (f) Extraction of the cylinder, (g) Recognition of two cylinders, (h) Extraction of four cylinders, (i) Restoration of a cuboid, (j) Overviews of the solid model of Example 2.

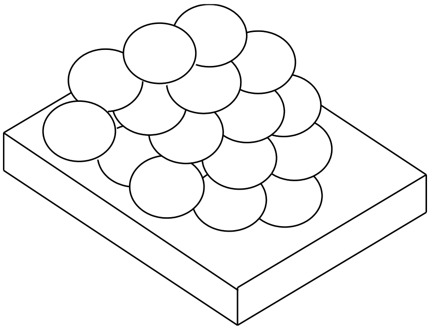


Fig. 11: The sketch of a bunch of grapes on a table.