20

2D Figure Pattern Mining

Keiji Gyohten, Hiroaki Kizu and Naomichi Sueda
Oita University,
Japan

1. Introduction

1.1 Background

With the recent enhancement of desktop design environments, it has become easy for personal users to design graphical documents such as posters, flyers, slides, drawings, etc. These kinds of documents are usually produced by the applications like drawing softwares, which have the advantage that they can store and retrieve the drawing data electronically. By reusing parts of the stored drawing data, the users can design the graphical documents much more easily.

However, generally, the stored data of many users is not shared, although this can be achieved by putting a drawing database. One reason is that it is difficult to retrieve desired figures from large amounts of drawing data in the database. Unlike in text search, the figure search will require enormous amounts of computation time because matching of the geometric primitives in the drawing data will cause their combinatorial explosion in 2D space.

To address this problem, many approaches have been proposed recently. When users search the drawing database, they should conjure up the desired figures and design their 2D sketches as the keys. In case of retrieving general figures, such as electrical symbols and map symbols, there would be little difference between sketches of them drawn by different users, but it is impractical to make them visualize various objects and things and use the sketches as the keys. For example, in case of retrieving human figures, since the sketches of humans differ according to the users, not all of figures of humans will be able to be obtained from the drawing database. To cope with this problem, we need a technique that enables the applications to automatically present users with the list of figures considered to have any meaning. Users can specify a figure of the desired object or thing simply by selecting it from the list and then retrieve the desired figures from the database using it as the key.

1.2 Figure pattern mining

Since the parts appearing many times in many graphical documents would illustrate something significant, the application should analyze all of the data in the drawing database and mine frequent figure patterns automatically so that it can obtain figures which represent some meanings. We call the frequent parts having similar figure patterns semantic figure patterns, which might well be the general figures like symbols. By presenting users with the list of the obtained semantic figure patterns, users can retrieve desired data from the drawing database without sketching. This will lead to sharing of the drawing data stored by many users.

In this chapter, we discuss a data mining approach for 2D drawing data designed by the drawing softwares and propose a method of 2D figure pattern mining. Usually, a variety of symbols registered in drawing softwares are placed on the drawing area through various kinds of 2D affine transformations such as rotation, scaling and translation. In order to mine such drawing data, including symbols placed in a complex manner, it should be represented in the form independent of these affine transformations. Since topology of geometric primitives is invariant to the affine transformations, our method first obtains interim results by mining topology data of the drawing data. After that, the final results are sorted out from the interim ones with the verification of their geometric validity.

In our approach, the topology data of the drawing data is represented by graphs where nodes and edges correspond to the geometric primitives and the spatial relations between them, respectively. We call this graph topology graph, where labels on the nodes denote the kinds of the geometric primitives and those on the edges the types of the spatial arrangements of two primitives. Even if the 2D affine transformation is applied to a partial set of the geometric primitives, its corresponding subgraph in the topology graph will be unchanged. Our method generates these topology graphs from all of the drawing data and mines them to extract the frequent subgraphs as the interim results. These subgraphs represent sets of the geometric primitives having frequent topological figure patterns, but the similarity of their appearance is not guaranteed.

Next, our method checks if the sets of the geometric primitives obtained in the previous step fit each other by inferring the affine transformations between them. If valid transformations can be estimated between the primitive sets, it can be said that they have the same topology and appearance. They can be interpreted as the final result, that is, which are the desired semantic figure patterns appearing frequently in the drawing database.

2. Related works

The analysis of figures has been studied by many researchers. In recent years, interesting study has been carried out not only on 2D figures but on 3D figures. However, the most important point is how to represent their shapes no matter what dimension they are in.

In the studies of the 3D shape matching, one general approach is to represent the 3D shapes with set of points. Barequet and Sharir presented a partial surface and volume matching method, which represents objects to be matched as a set of points and estimates a transformation of one object to the other (Barequet & Sharir, 1997). Aiger et al. proposed a fast and robust alignment scheme for surface registration of 3D point sets (Aiger et al., 2008). Some approaches dealt with not set of points but salient points and achieved advanced matching like partial matching (Novotni et al., 2005) and matching over several views of an object (Castellani et al., 2008).

There is also an approach where the 3D shapes are represented with skeleton graphs. Brennecke and Isenberg used the skeleton graph to calculate a similarity measure for 3D geometry models (Brennecke & Isenberg, 2004). Sundar et al. used graph matching techniques and proposed a method of part-matching of 3D objects. This method is intended to be used for retrieval of the shapes from an indexed database (Sundar et al., 2003). Tung and Schmitt proposed the multiresolution Reeb graph and tried to improve a shape matching method applied to content-based search in database of 3D objects (Tung & Schmitt, 2005). Iyer et al. proposed a shape representation which has multiple levels of detail and preserves geometry and topology of 3D models using a hierarchical skeleton graph.

This is also used for similar 3D shape retrieval. Schnabel et al. used not only graph representation but 3D point-clouds and presented a flexible framework for the rapid detection of semantic shapes (Schnabel et al., 2007).
In order to develop a fast retrieval system of 3D objects, it is important to represent features of the shapes simply and plainly for the reduction of the computation time (Bustos et al., 2004). In this case, many methods retrieve desired objects without considering detailed correspondence between 3D shapes and exploit only the 3D shape features like Krawtchouk moment (Mademlis et al., 2006), Bag-of-Words representation (Li & Godil, 2009) and Bag-of-features SIFT (Ohbuchi et al., 2008).
The 2D shape matching technique has been studied actively in the field of data retrieval where 2D shape is used as a search key. Kim and Grauman presented an asymmetric region-to-image matching method which identifies corresponding points for each region and compares images by considering geometric consistency and appearance similarity (Kim & Grauman, 2010). Liu et al. proposed a sketch-based approach to find matching source images for image composition. The system asks users to draws a rough sketch to identify the desired object and finds a set of matching images (Liu et al., 2009). This sketch-based approach is also used in searching 3D objects. Funkhouser et al. proposed a shape descriptor for boundary contours and used it as the shape-based query. This descriptor is invariant to rotation and is represented with a set of the amplitudes of constituent trigonometrics (Funkhouser et al., 2003). Pu et al. proposed 2D sketch-based user interface for 3D CAD model retrieval, where 2D shapes are compared with each other by matching a large amount of sample points on their edges and calculating the Euclidean distance distribution between the sample points (Pu et al., 2005).
In all cases, most of the techniques for analyzing shapes of figures have been used mainly for matching and data retrieval. There does not appear to be any data mining methods in this field. However, some interesting approaches have been proposed. They analyze shapes without any prior knowledge about the desired objects. Lovett et al. proposed an incremental learning technique for the generalization of object categories based upon the sketches of those objects. The generalized categories are used to classify new sketches (Lovett et al., 2007). Hou et al. proposed a clustering method based on Support Vector Machines to organize the 3D models semantically. The resultant clusters are used to classify the unknown data (Hou et al., 2005). Pauly et al. presented an approach for discovering regular or repeated geometric structures in 3D shapes, which are represented in point or mesh based models (Pauly et al., 2008). Ovsjanikov et al. proposed an approach for computing intrinsic symmetries of a 3D shape (Ovsjanikov et al., 2008).
Since figures are drawn in a multi-dimensional space, it can be said that a shape of a figure is a kind of spatial data. In the field of data mining, spatial data mining is becoming popular and has been studied by many researchers recently (Ng and Han, 1994). Sheikholeslami et al. proposed a multi-resolution clustering method which can effectively identify arbitrary shape clusters at different degrees of accuracy in spatial databases using wavelet transformation (Sheikholeslami et al., 1998). Jiang proposed a spatial hierarchical clustering technique for generalization processes in GIS (Jiang 2004). Visual data mining proposed by Brecheisen et al. is a very interesting approach, where the hierarchical clustering structure of a 3D object database is visualized (Brecheisen et al., 2004).

3. Topology graph mining

Our method represents spatial relations between the geometric primitives as the topology graph, which is invariant to the 2D affine transformations such as rotation, scaling and translation. Sets of the geometric primitives having frequent topological figure patterns can be obtained by generating the topology graphs from large amounts of 2D drawing data and applying a graph mining method to them. The desired semantic figure patterns will be included in these geometric primitive sets.

3.1 Topology graph

In the topology graph, a node corresponds to a geometric primitive, which is denoted as f_h. There exists an edge between the nodes if their corresponding geometric primitives, denoted as (f_h, f_k), touch or intersect with each other. From the above, the topology graph can be denoted as $\mathbf{G}=(\{f_h\}, \{(f_h, f_k)\})$.

Now we consider the topology of the geometric primitives and define the labels on the nodes and edges to deal with their spatial arrangements. Since each geometric primitive in the 2D drawing data generally has its type (line segment, circle, etc.) and the control points (start point, end point, center, etc.) specifying its shape, it can be expressed as $(t^{fh}, \{\mathbf{p}^{fh}{}_i\})$, where t^{fh} and $\{\mathbf{p}^{fh}{}_i\}$ is the type and the set of control points of the geometric primitive f_h, respectively. It should be noted that the coordinates of the control points vary according to the 2D affine transformation applied to the geometric primitives. Since the topology graph should be invariant to the 2D affine transformation, the node labels show only the types of the geometric primitives with a digit as shown in Fig. 1. The control points will be considered for the analysis of the spatial relations between the geometric primitives and for the estimation of the affine transformations between the sets of the geometric primitives, as described later.

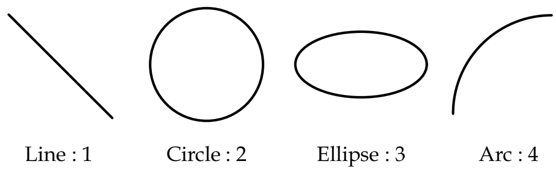

Fig. 1. Digits assigned to geometric primitives

The edge labels represent the spatial relations between the geometric primitives touching or intersecting each other. The spatial relations must be able to be defined for any types of geometric primitives. Therefore, we suppose that each geometric primitive consists of three components: area, boundary and set of end points as shown in Fig. 2 and define five digits as the edge label by considering form of overlapping between these components as shown in Fig. 3. The first digit represents spatial relation between areas of geometric primitives. If these areas overlap partially, the first digit is set to be 1. If an area is entirely included in the other, it is set to be 2. The second digit represents the relationship between the boundaries. It is the number of their intersections. If there is infinite set of the intersections, let the second digit be 9. The third digit is the number of end points which are included in the area of the other geometric primitive. This represents the relationships between the area and the end points. The fourth digit is the number of the end points which are on the boundary of the other geometric primitive. This represents the relationships between the boundary and the end points. The fifth digit is the number of end points that are shared with the other geometric primitive. This represents relationships between the end points. Examples of the edge labels are shown in Fig. 4. These edge labels are unchanged even if their relevant

geometric primitives are moved by the affine transformation. Naturally, the ways of calculating the edge labels differ according to the combination of the geometric primitive types. The an edge label calculation between two line segments is totally different from that between an line segment and a circle. Our method assumes that all of the ways of the edge label calculation are given for any combination of possible geometric primitive types.
Our method generates the topology graphs for all 2D drawing data in the drawing database to represent topological figure patterns. The node labels, the edge labels, and the connections between the nodes are invariant to the 2D affine transformations.

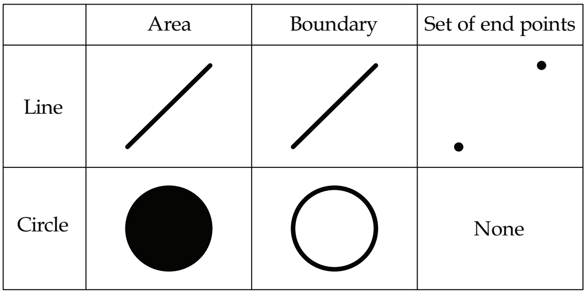

Fig. 2. Area, Boundary and Set of end points of geometric primitives

	Area	Boundary	Set of end points
Area	1		3
Boundary		2	4
Set of end points			5

Fig. 3. Places of digits for edge label

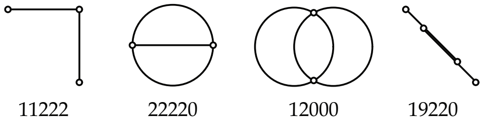

Fig. 4. Examples of edge label

3.2 Mining of topology graph

By mining the topology graphs, our method extracts subgraphs corresponding to the frequent topological figure patterns, which could be the desired semantic figure patterns. In the field of graph mining, various methods have been proposed to obtain frequent subgraphs fast and correctly. Our method uses GASTON graph mining algorithm, which works efficiently, especially for sparse undirected graphs (Nijssen & Kok, 2004). Since the edges in the topology graphs correspond to spatial relations between geometric primitives adjacent to each other, the number of the edges is generally much less than the possible number of edges. Therefore, GASTON will works efficiently for the topology graphs.
Although the extracted subgraphs represent the frequent topological figure patterns, their appearances are not always the same. As shown in Fig. 5, the same topology subgraphs do not always show the same configurations of the geometric primitives. Our method overlays

one of the extracted sets of the geometric primitives onto the other to check the similarity of their appearances. This is done by verifying that a valid 2D affine transformation can be obtained between the primitive sets.

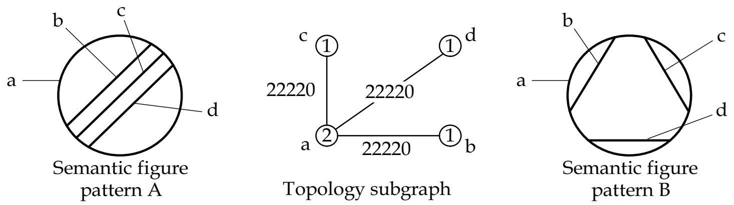

Fig. 5. Same topology subgraph of different semantic figure patterns

4. Check of geometric validity

In order to verify the geometric validity of the sets of the geometric primitives obtained by the topology graph mining, our method tries to overlay one of the sets onto the other and checks the similarity of their appearances. This can be done by taking the following two steps: First, the 2D affine transformation is estimated between the primitive sets. Then their appearances are compared by fitting them with each other using the estimated transformation. If it is judged that they have the same appearance, they can be considered as the final results, that is, frequent figure patterns representing the desired semantic figure patterns.

If coordinates of 2D points are represented as augmented vectors, a 2D affine transform can be expressed as follows:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & d_1 \\ a_2 & b_2 & d_2 \\ 0 & 0 & e \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{1}$$

where $[x\ y\ 1]^t$ and $[x'\ y'\ 1]^t$ are the coordinates of an original point and of the point after the transformation in the homogeneous coordinate system. a_1, a_2, b_1, b_2, d_1, d_2 are the affine parameters. e is a parameter due to the use of augmented vectors. If n correspondences of the points between two sets of the geometric primitives are obtained by topology graph mining, the following equation is derived:

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1' \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y_1' \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2' \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -y_2' \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & - \\ x_n & y_n & 1 & 0 & 0 & 0 & -x_n' \\ 0 & 0 & 0 & x_n & y_n & 1 & -y_n' \end{bmatrix} \begin{bmatrix} a_1 \\ b_1 \\ d_1 \\ a_2 \\ b_2 \\ d_2 \\ e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2}$$

where $\mathbf{p}_i=[x_i\ y_i\ 1]^t$ and $\mathbf{p}_i'=[x_i'\ y_i'\ 1]^t$ (i=1, ..., n) are the coordinates of n corresponding points. Eq. (2) can be represented as follows:

$$R_a = 0 \ . \tag{3}$$

In practice, Eq. (3) is usually an over-determined system of equations because the number of the control point coordinates is usually more than the number of unknown parameters, that is, affine parameters and e.
Our method obtains the least square solution of Eq.(3) using the Lagrange multiplier method. The value to be minimized is the sum square of the residuals between the corresponding control points as shown in Fig. 6. This value can be represented as follows:

$$C = a^t R^t R a \,. \tag{4}$$

Here one solution to Eq.(4) is **a** = **0**, which is meaningless for the purpose of this approach. To avoid this problem, we set the constraint

$$a^t a = 1 \tag{5}$$

on the unknown parameters and set up the objective function as follows:

$$C = a^t R^t R a + \lambda(a^t a - 1)\,, \tag{6}$$

where λ is a Lagrange multiplier. Since the partial differentiation of Eq.(6) is

$$\frac{\partial C}{\partial a} = 2R^t R a + 2\lambda a\,, \tag{7}$$

the desired solution is the eigenvector of $\mathbf{R}^t\mathbf{R}$ which corresponds to the minimum eigenvalue.
It should be noted that our method cannot determine a unique correspondence between the control points in the frequent topological figure patterns obtained by the topology graph mining as shown in Fig. 7. Our method obtains all of the combination, estimates affine parameters for each of them, and selects the most valid affine transformation. As shown in Fig. 8, we denote by $\mathbf{S}^F$ the sets of all possible combination of geometric primitives obtained from the results of the topology graph mining. For a geometric primitive combination g $\in$ $\mathbf{S}^F$, the sets of combination of control points are denoted by $\mathbf{S}^P{}_g$. Our method minimizes the standard deviation of the residuals between control points corresponding with each other as follows:

$$d = \min_{g \in S^F} \min_{c \in S_g^P|} \left(sd(\{ | t(p_i^c, a^c) - p_i^c{}' | \}) \right), \tag{8}$$

where sd({v}) is the standard deviation of the values {v}. **t**(**p**, **a**) is the point translated from the point **p** through the affine transformation whose parameters are **a**. Points $\mathbf{p}_i{}^c$ and $\mathbf{p}_i{}^{c'}$ are the i-th corresponding points in the control point combination c. $\mathbf{a}^c$ is the estimated parameters in the combination c using the Lagrange multiplier method as described above. If d is smaller than a threshold, we can state that the set of geometric primitives evaluated with (6) have the same topology and that one of them can be mapped to the other through an affine transformation as shown in Fig. 6. This implies that they represent the same semantic figure pattern.
In this appearance evaluation process, thresholding d starts from the pairs of geometric primitive sets corresponding to the largest topology subgraph. If they are judged to have the same semantic figure pattern, the geometric primitives in them are excluded from this process to avoid extracting their substructures as other semantic figure patterns.

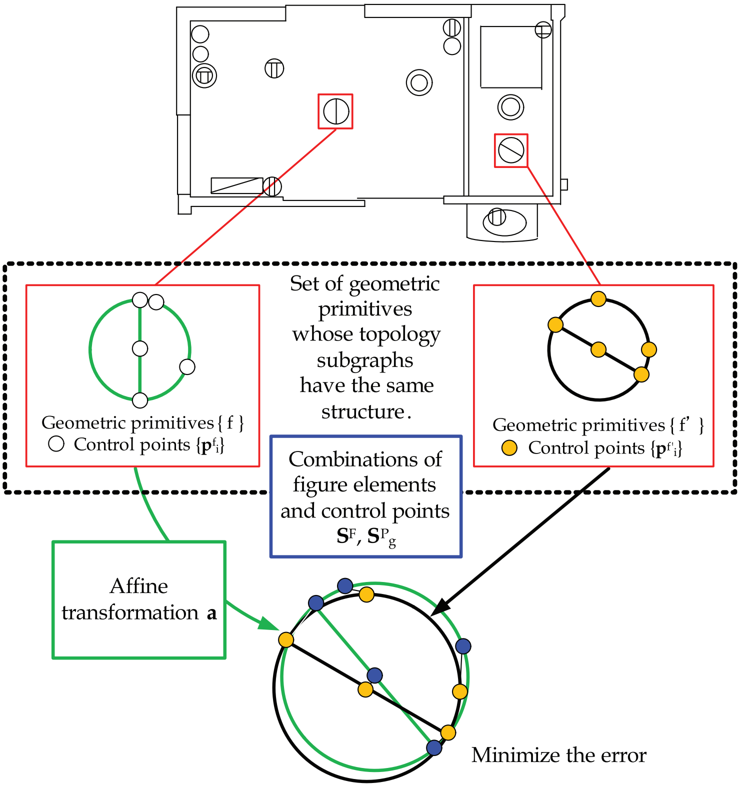

Fig. 6. Estimation of affine transformation

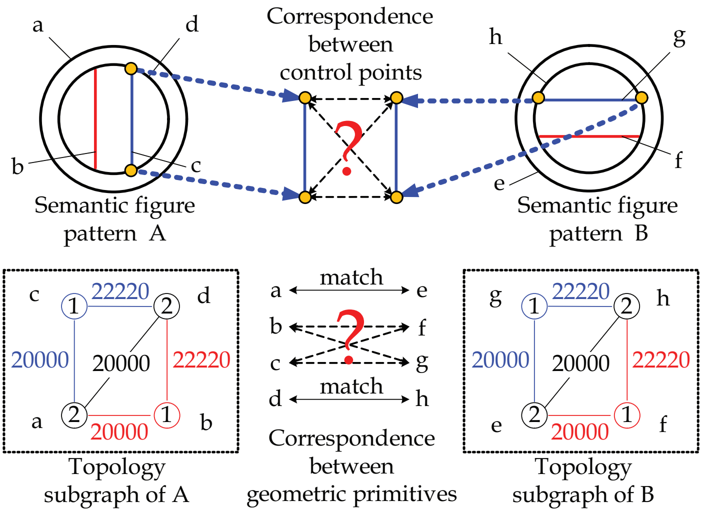

Fig. 7. Ambiguity of correspondence between semantic figure patterns

5. Experiments

We implemented the proposed method on a PC with Intel Core 2 Duo CPU, 2.13GHz, and 4GB RAM using the C++ language and applied it to some 2D drawing data to confirm its validity. As the 2D drawing data for this experiment, we used CAD data of 42 floor plans where there are 8 electrical symbols, which are shown in Fig. 8. These floor plans were drawn by 6 users with Microsoft Office Visio.

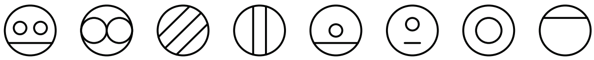

Fig. 8. Electrical symbols

In the experiment, we evaluated precision and recall to evaluate the performance of the proposed method. They are computed as follows:

$$\mathrm{Precision} = \frac{N_{tp}}{N_{tp} + N_{fp}}, \tag{9}$$

$$\mathrm{Recall} = \frac{N_{tp}}{N_{tp} + N_{fn}}, \tag{10}$$

where N_{tp} is the number of the semantic figure patterns extracted correctly. N_{fp} is the number of the extracted figure patterns which are not true semantic figure patterns. N_{fn} is the number of the semantic figure patterns not extracted in this experiment.

First, we built topology graphs of all floor plans and applied the topology graph mining to them with the support 1.0. Here the support is defined as follows:

$$\mathrm{Support} = \frac{M_{ff}}{M_{af}}, \tag{11}$$

where M_{ff} is the number of the floor plans where the frequent topology subgraphs appear, and M_{af} is the total number of floor plans. This value can adjust the sensitivity of the mining performance. As the result of the topology graph mining, our method extracts 12 kinds of topology subgraphs, which represent the frequent topological figure patterns and potentially desired semantic figure patterns as illustrated in Fig. 9. These frequent topological figure patterns included 5 out of 8 types of electrical symbols. The remaining 3 types could not be obtained because of the following reasons. First, some of the types did not appear frequently enough to satisfy the support value. The other reason is the error of labelling some edges, where, for example, a geometric primitive did not quite reach to the other one, even though they should touch each other, and vice versa as shown in Fig. 10. In this step, the precision and recall were 66% and 82%, respectively.

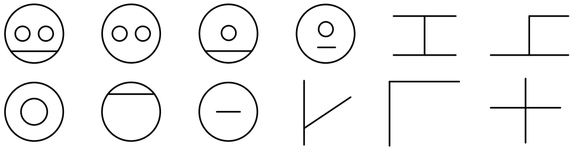

Fig. 9. Result of topology graph mining

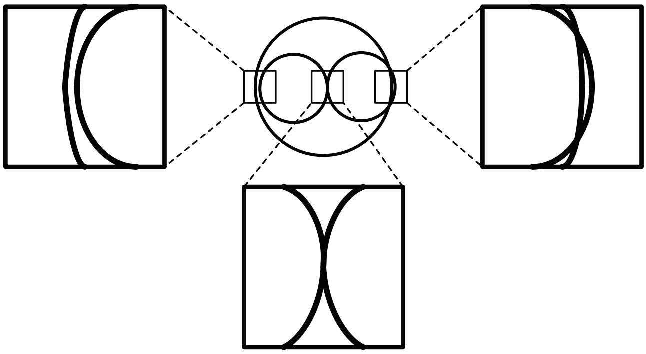

Fig. 10. Example of labeling error

Next, we eliminated wrong semantic figure patterns by checking the appearance similarity between the geometric primitive sets corresponding to the same topological subgraphs. The threshold for d was set to be 1.0 which was determined experimentally. In this step, the precision was improved to be 72%. However, the proposed method has to estimate affine transformations for all of the combinations of geometric primitives and those of control points individually. This will lead to the explosion of the computation time. Moreover, not all of the erroneous semantic figure patterns were eliminated in this step because similar configurations could exist among them. Figure 11 shows an example, where the geometric primitives are arranged similarly, even though they do not represent electrical symbols.

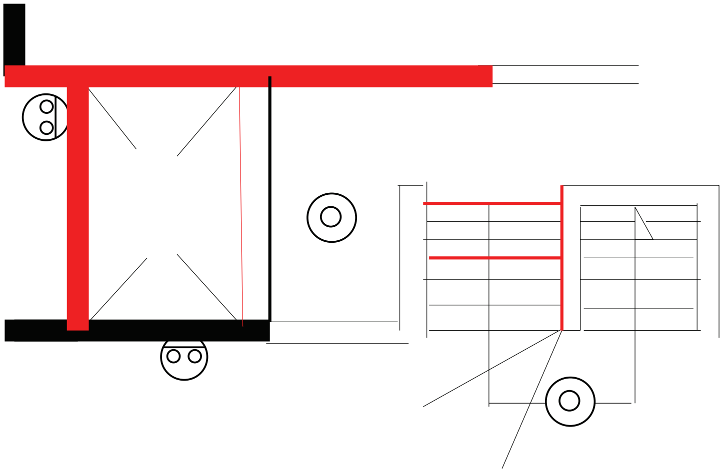

Fig. 11. Example of erroneous figure patterns having similar configuration

In the end, the proposed method shows the obtained results and asks the user to choose correct semantic figure patterns using the GUI shown in Fig. 12. In this step, wrong semantic figure patterns were greatly reduced and the precision rose to 92%. This work is light because the user only selects correct semantic figure patterns. Figura 13 shows one of the resultant floor plans. Table 1 lists the precision and recall at each step.

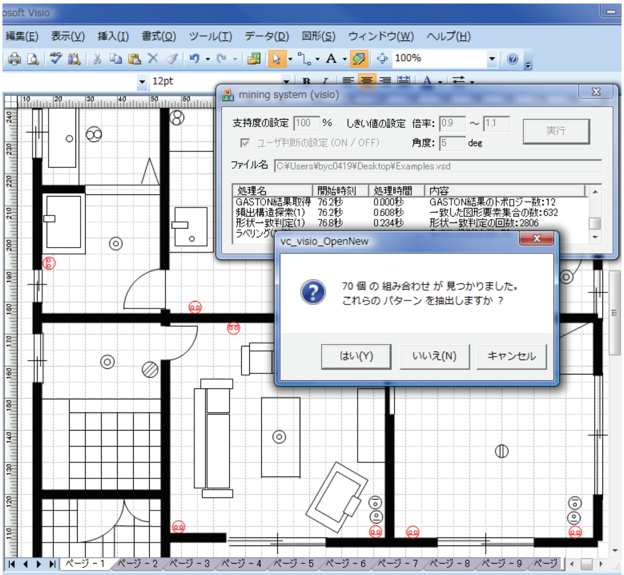

Fig. 12. GUI

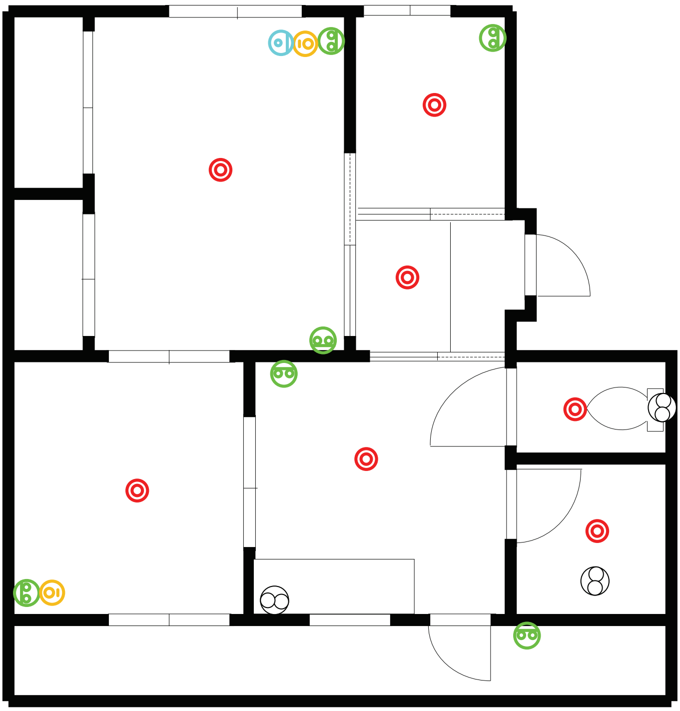

Fig. 13. Example of result

	Precision	Recall
Topology graph mining	66%	82%
Topology graph mining and affine transformation	72%	82%
Topology graph mining and affine transformation and interaction with user	92%	82%

Table 1. Precision and recall at each step

6. Discussion

We now consider the experimental results and future development. One of the serious problems is the explosion of the computation time, especially in the case of dealing with large amounts of the drawing data. In our method, the reduction of the computation time is equivalent to reducing the combinations of geometric primitives and those of control points in the estimation of the affine transformations. This problem is caused by the use of undirected graphs as the topology graphs, which gives rise to the ambiguity of the correspondence between the geometric primitives and that between the control points. This ambiguity increases the combinations of geometric primitives and those of control points. For example, in the case of Fig. 14(a), the proposed method generates the same topology graphs and confuses the correspondence between the geometric primitives and that between the control points. If we can use the directed graphs as the topology graphs, the results obtained in the topology mining step will make the correspondences clear as illustrated in Fig. 14(b). This will lead to a reduction in the number of combinations and the amount of computation time. To actualize this approach, we should develop and use a fast graph mining method for directed graphs.

The failure of the extraction of the desired semantic figure patterns is caused by the change of edge labels that occurs when the corresponding geometric primitives are displaced slightly as illustrated in Fig. 10. Since this failure will lead to poor recall, this is the problem to be solved at present. But this problem is very difficult and leaves room for future studies. The basic approach to this problem is to simplify the description of the edge labels so that their drastic change does not occur in the case of a slight displacement of the geometric primitives. Contrary to this, the precision can be improved using knowledge on the desired object. In the case of dealing with the floor plans, for example, the system can assume that the desired symbols consist of geometric primitives in a large circle. But this will compromise the generality of the proposed method. From this viewpoint, we believe that the objects extracted erroneously should be excluded with the user interaction.

This time we used the drawing data made with Microsoft Office Visio. If the proposed method becomes applicable to the data made with Microsoft Office PowerPoint, the drawing data distributed on the Internet could be sorted and subsequently exploited by many users by incorporating social tagging into the figure pattern mining method (Setz & Snoek, 2009). This is a kind of automatic clipart generation.

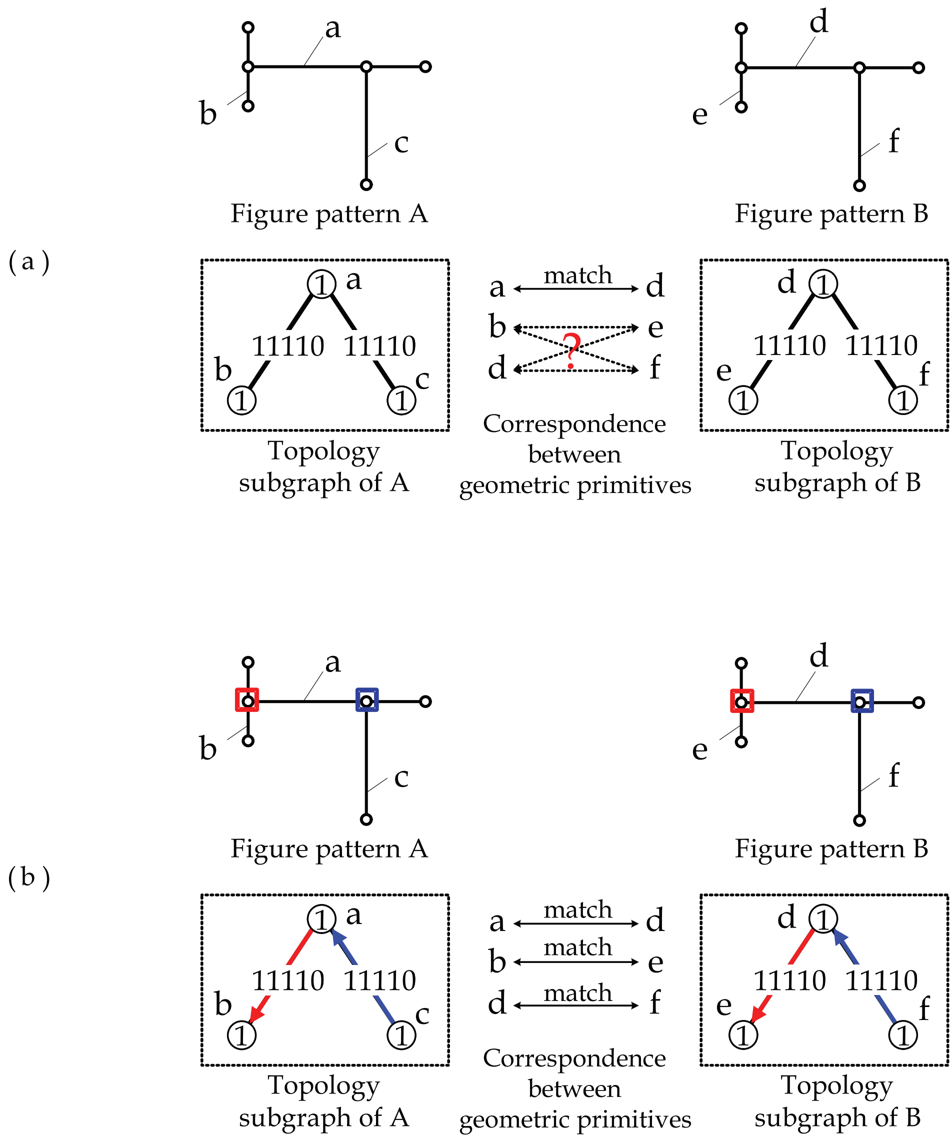

Fig. 14. Undirected and directed topology subgraphs

7. Conclusion

In this chapter, we described a 2D figure pattern mining approach where semantic figure patterns can be obtained from the drawing data without prior knowledge. The proposed method first builds the topology graphs to represent topology of geometric primitives in the drawing data. Then our method extracts frequent topology subgraphs by mining all of the

topology graphs and tries to sort out correct semantic figure patterns from them by inferring affine transformations among the sets of their corresponding geometric primitives.
In the experiment, 82% of electrical symbols placed in floor plans could be extracted through the interaction with the user. However, some electrical symbols were not extracted in the cases where the electrical symbols were placed in few floor plans and where an edge label was changed by the slight error of geometric primitive positions. We hope this kind of study will continue along the lines described in the previous section.

8. References

Aiger, D.; Mitra, N. J. & Cohen-Or, D. (2008). 4-Points Congruent Sets for Robust Pairwise Surface Registration, *ACM Transactions on Graphics*, Vol. 27, No. 3, August 2008, pp. 1-10

Barequet, G. & Sharir, M. (1997). Partial Surface and Volume Matching in Three Dimensions, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 9, September 1997, pp. 929-948

Brecheisen, S.; Kriegel, H.; Kroger, P. & Pfeifle, M. (2004). Visually Mining Through Cluster Hierarchies, *Proceedings of the 4th SIAM International Conference on Data Mining*, April 2004, pp. 400-412

Brennecke, A. & Isenberg, T. (2004). 3D Shape Matching Using Skeleton Graphs, In: *Simulation and Visualisierung*, pp. 299-310, SCS Europen Publishing House

Bustos, B.; Keim, D.; Saupe, D.; Schreck, T. & Vranic, D. (2004). An Experimental Comparison of Feature-Based 3D Retrieval Methods, *Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization, and Transmission*, September 2004, pp. 215-222

Castellani, U.; Cristani, M.; Fantoni, S. & Murino, V. (2008). Sparse points matching by combining 3D mesh saliency with statistical descriptors, *Computer Graphics Forum* Vol. 27 No. 2, April 2008, pp. 643-652

Funkhouser, T.; Min, P.; Kazhdan, M.; Chen, J.; Halderman, A.; Dobkin, D. & Jacobs, D. (2003). A search engine for 3D models, *ACM Transactions on Graphics*, Vol. 22, No. 1, January 2003, pp. 83-105

Hou, S.; Lou,K. & Raman, K. (2005). SVM-based Semantic Clustering and Retrieval of a 3D Model Database, *Computer-Aided Design & Applications*, Vol. 2, 2005, pp. 155-164

Iyer, N.; Jayanti, S.; Lou, K.; Kalyanaraman, Y. & Ramani, K. (2004). A Multi-Scale Hierarchical 3D Shape Representation for Similar Shape Retrieval, *Proceedings of the International Symposium on Tools and Methods for Competitive Engineering*, April 2004, pp. 1117-1118

Jiang, B. (2004). Spatial Clustering for Mining Knowledge in Support of Generalization Processes in GIS, *ICA Workshop on Generalisation and Multiple representation*, August 2004

Kim, J. & Grauman, K. (2010). Asymmetric Region-to-Image Matching for Comparing Images with Generic Object Categories, *Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition*, June 2010

Li, X. & Godil, A. (2009). Exploring the Bag-of-Words method for 3D shape retrieval, *Proceedings of the 16th IEEE International Conference on Image Processing*, November 2009, pp.437-440

Liu, Y.; Xu, D.; Wang, J.; Tang, C. K.; Huang, H.; Tong, X. & Guo B. (2009). Sketch and Match: Scene Montage Using a Huge Image Collection, *Tech. Report* MSR-TR-2009-134 Microsoft, September 2009

Lovett, A.; Dehghani, M. & Forbus, K. (2007). Incremental Learning of Perceptual Categories for Open-Domain Sketch Recognition, *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, January 2007, pp. 447-452

Mademlis, A.; Axenopoulos, A.; Daras, P. Tzovaras, D. & Strintzis, M. G. (2006). 3D Content-Based Search Based on 3D Krawtchouk Moments, *Proceedings of the Third International Symposium on 3D Data Processing*, Visualization, and Transmission, June 2006, pp.743-749

Ng, R. T. & Han, J. (1994). Efficient and Effective Clustering Methods for Spatial Data Mining, *Proceedings of the 20th International Conference on Very Large Data Bases*, September 1994, pp. 144-155

Nijssen, S. & Kok, J. (2004). A Quickstart in Frequent Structure Mining Can Make A Difference, *Proceeding of the 2004 ACM SIGKDD International Conference on Knowledge Discovery in Databases*, August 2004, pp. 647-652

Novotni, M.; Degener, P. & Klein, R. (2005). Correspondence Generation and Matching of 3D Shape Subparts, *Technical Reports* CG-2005-2 Universitat Bonn, ISSN 1610-8892

Ohbuchi, R.; Osada, K.; Furuya, T. & Banno, T. (2008). Salient Local Visual Features for Shape-Based 3D Model Retrieval, *Proceedings of IEEE International Conference on Shape Modeling and Applications*, June 2008

Ovsjanikov, M., Sun, J. & Guibas, L. (2008). Global Intrinsic Symmetries of Shapes, *Computer Graphics Forum* (Eurographics Symposium on Geometry Processing), Vol.27, No.5 July 2008, pp. 1341-1348

Pauly, M.; Mitra, N. J.; Wallner, J.; Pottmann, H. & Guibas,L. (2008). Discovering Structural Regularity in 3D Geometry, *ACM Transactions on Graphics*, Vol. 27, No. 3, August 2008, pp.1-11

Pu, J.; Lou, K. & Ramani, K. (2005). A 2D Sketch-Based User Interface for 3D CAD Model Retrieval, *Computer-Aided Design & Applications,* Vol. 2, No. 6, 2005, pp. 717-725

Schnabel, R.; Wahl, R.; Wessel, R. & Klein, R. (2007). Shape Recognition in 3D Point Clouds, *Technical Reports* CG-2007-1 Universitat Bonn, ISSN 1610-8892

Setz, A.T. & Snoek, C.G.M. (2009). Can Social Tagged Images Aid Concept-Based Video Search? *Proceedings of IEEE International Conference on Multimedia and Expo*, June 2009, pp. 1460-1463

Sheikholeslami, G.; Chatterjee, S. & Zhang, A. (1998). WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases, *Proceedings of the 24rd International Conference on Very Large Data Bases,* August 1998, pp. 428-439

Sundar, H.; Silver, D.; Gagvani, N. & Dickinson, S. (2003). Skeleton Based Shape Matching and Retrieval, *Proceedings of Shape Modeling International,* May 2003, pp.130-139

Tung, T. & Schmitt, F. (2005). The augmented multiresolution Reeb graph approach for content-based retrieval of 3D shapes, *International Journal of Shape Modeling*, Vol. 11, No. 1, June 2005, pp. 91-120