

Attributed Graph Matching Based Engineering Drawings Retrieval

Rujie Liu¹, Takayuki Baba², and Daiki Masumoto²

¹ Fujitsu Research and Development Center Co. LTD, Beijing, P.R.China

² Information Technology Media Labs, Fujitsu Laboratories LTD., Kawasaki, Japan
rjliu@frdc.fujitsu.com

Abstract. This paper presents a method for engineering drawings retrieval by their shape appearances. In this method, an engineering drawing is represented by an attributed graph, where each node corresponds to a meaningful primitive extracted from the original drawing image. This representation, which characterizes the primitives as well as their spatial relationships by graph nodes attributes and edges attributes respectively, provides a global vision of the drawings. Thus, the retrieval problem can be formulated as one of attributed graph matching, which is realized by mean field theory in this paper. The effectiveness of this method is verified by experiments.

Keywords: Engineering drawings retrieval, Attributed graph, Mean field theory

1 Introduction

On-line maintenance of large volume of documents, such as engineering drawings, has become a major research area recently due to the ever increasing rate at which these documents are generated in many application fields [1, 2]. It is very common a thing for designers and draft people to refer to previous drawing documents to get some inspiration or solutions already achieved. However, retrieving these documents generally is a slow and tedious work, which requires an exhaustive examination of the whole engineering drawing database. To facilitate such retrieval, textual content such as keywords has been widely used. While this information is helpful in retrieval, it is, however, a heavy work to generate such descriptions manually, besides this, several keywords only are always incapable of describing the true content in a drawing.

Technical drawings retrieval, or any other type of matching problem, can be seen as a correspondence calculation process. Whether a database drawing is retrieved or not is thus determined by the correspondence value between the query drawing and this database drawing. For this purpose, the technical drawings are first represented by attributed graphs, where graph nodes correspond to meaningful primitives extracted from the original drawing image, such as lines and curves, while the spatial relationships between these primitives are described by graph edges. Next, the graph nodes in query drawing are regarded as a set of labels to label those nodes in a database drawing so as to determine the correspondence between them. With this manner, both the database drawings which are similar to the query one and those which include a similar part with the query one can be obtained.

To generate attributed graphs from engineering drawing images, the original raster images should be firstly converted to vector form, say, vectorization. As a preliminary

step of document analysis and processing, many vectorization techniques have been developed in various domains [3-5]. In this paper, it is assumed that the affiliated information such as annotation text has been removed, and all the curves in the drawing images have been converted into one-pixel-width by some thinning algorithm. Firstly, the drawing image is decomposed into rough primitives based on the intersection points. Then, a merge-split process is used to make them more meaningful.

The primitives obtained by above process are treated as nodes to build attributed graph. Different from Beniot's N-Nearest neighbor method [6], the Delaunay tessellation strategy is used to generate the structural description of engineering drawings. With this manner, the spatial relationships between these primitives are represented more naturally and meaningfully.

By above process, the content and the structure of engineering drawings are represented by means of attributed graphs. This enables us to interpret the retrieval problem as one of matching two attributed graphs. In recent years, some methods have been proposed to solve graph matching problem [7-11]. Based on the compromise between speed and performance, mean field theory is adopted in this paper to solve graph matching problem.

The remaining of this paper is arranged as follows: In section 2, we describe the engineering drawing decomposition process to divide the drawing image into meaningful primitives. How to construct graphs from these primitives, as well as attributes definition, is introduced in section 3. In section 4, graph matching algorithm is outlined. At last, the experiments as well as some discussion are given in section 5.

2 Primitives Extraction

Engineering drawings are mainly consisted of basic primitives, which are assembled together by specific spatial distribution and structural relationship into cognitively meaningful objects. This hints us that a structural description can be employed to well represent the content of an engineering drawing. In this paper, a drawing image is firstly decomposed into rough primitives, which are then refined by a merge-split process into meaningful form.

2.1 Engineering Drawing Image Decomposition

As mentioned in the introduction, it is assumed that some pre-processing has been applied to original drawing images to remove the affiliated information such as annotation text, make them binary and composed of one-pixel-width curves only. For this kind of image, a simple method is designed to decompose it into rough primitives:

Rough primitives extraction is implemented by following strategy:

- 1) Calculate the number of 8-connected neighbors for each pixel.
- 2) Link these pixels together, which are 8-connected and have no more than two neighbor pixels, to form a rough primitive.

By this process, a curve maybe divided into several fractions, which should be merged by some measures to form a meaningful one. Following, a merge-split process will be used to re-arrange these rough primitives into meaning form.

2.2 Merge

The objective of merge is to recover original drawing elements, such as curves, straight lines, from the obtained rough primitives. As shown in figure 1 where three curves intersect together, 6 rough primitives are obtained from above decomposition process, it is hoped that the original three curves (lines) can be recovered by this merging process.

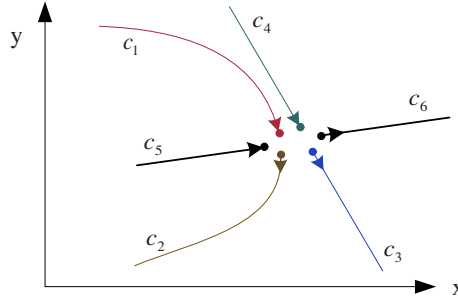


Fig. 1. Illustration of merging operation

It is a natural thing to adopt tangent direction and spatial distance as the basis of merging operation. If two rough primitives are collinear and close to each other at their end points, there exists possibility for these two primitives to be merged together. Besides this, to get a reasonable result, all the rough primitives near the intersection must be considered simultaneously when determining the merge operation.

Based on above idea, following algorithm is designed to determine merging operations.

➤ Criterion

For a pair of rough primitives, following criteria must be satisfied if they are to be merged:

- 1) Merge operation can only be performed at the end points rough primitives.
- 2) Exclusivity. Merging is allowed at most once at an end point of a rough primitive.
- 3) Collinearity and space gap. Only the tangent direction and the spatial difference between two rough primitives' end points is small than a threshold, can they be merged at these two end points.

Above criteria rule out most of rough primitives when considering possible merging for one rough primitive. However, several candidates are still survived, from which the most reasonable should be determined. For example, in figure 1, segments c_4 to c_6 are ruled out from c_1 's candidates due to collinearity criterion, in other words, they can't be merged with c_1 , however, both c_2 and c_3 are still possible to be merged with c_1 . In order for the exclusivity criterion to be satisfied, the most reasonable rough primitive, c_2 in this example, should be selected from the candidates.

The merging operation orderly traverses all rough primitives and processes each of the two end points of every rough primitive respectively. Assume c_i the rough primitive being processed, the steps of merging operation are as follows:

- (1) Select an end point of c_i and denote it as a .
- (2) Exclusivity checking. Check whether c_i has been merged at the end point a , if so go to step (7), otherwise go on the next step.
- (3) Obtain the merging candidates set of c_i with space gap and collinearity rules. Check other rough primitives outputted from the image decomposition part, record the rough primitives and their corresponding end points that meet the following these rules.

Denote $c_i^m, m=1, \dots, M$ as the candidate set, $d_i^m, m=1, \dots, M$ the tangent direction difference between the corresponding end points of these rough primitives and end point a of c_i .

- (4) Calculate the minimum value of d_i^m , $\hat{d}_i = \min\{d_i^m, m=1, \dots, M\}$.
- (5) Set the *tolerance* value and further narrow the range of candidate set:

$$\hat{c}_i^m = \left\{ c_i^m \mid d_i^m < \hat{d}_i + \textit{tolerance} \right\}$$
 \hat{c}_i^m represents the new candidate set.
- (6) If the size of \hat{c}_i^m equals to 1, say, there is only one element in set \hat{c}_i^m , denoted as \hat{c}_j . Regard \hat{c}_j as the current rough primitive and calculate \hat{c}_j 's candidate set at its corresponding end point (recorded in step (3)) with steps (3) to (5). If the candidate set of \hat{c}_j only contains rough primitive c_i and the corresponding end point of c_i is a , merge them together at \hat{c}_j 's corresponding end point and end point a of c_i .
- (7) Denote the other end point of c_i as b , repeat above steps (2)-(6) similar to end point a .
- (8) Check whether all rough primitives have been traversed, if not, select the next rough primitive as current one and go to step (1); otherwise go to step (9).
- (9) Check whether any merge operation occurred during this traverse, if not, finish the program, otherwise repeat steps (1)-(8) to perform next traverse.

Unlike continuous curve, it is nearly impossible to get the true tangent direction value of discrete curve. Therefore, some errors will be resulted if only a simple minimum direction difference strategy is adopted. To cope with such cases, a tolerance value is introduced in above method to prevent the loss of true rough primitive from candidate set.

2.3 Split

In the rough primitives obtained from above process, there may exist sharp bends. Here, a split process is adopted to divide a primitive with sharp bends into several smooth ones.

Based on the curvature values, the significant bending points in each primitive can be found, which are then used to split this primitive into several smooth ones.

- 1) Bending points detection. Let the primitive pixel and its curvature be c_t and k_t , c_t is deemed as bending points if following conditions are satisfied: (a). k_t is local maximum(minimum) in a k_t centered window of width w . (b). $k_t > \alpha * k_{max}$, where k_{max} is the maximum curvature value. (c). $|k_t| > \beta$, where β is a predefined positive threshold.
- 2) Remove a bending point if it is not obvious. Let p_1 and p_2 be two consecutive bending points, c_1, c_2, c_{avg} the curvatures corresponding to p_1, p_2 and the average curvature between these two points respectively. If: $c_{avg} > \delta * (c_1 + c_2) / 2$, where δ is a predefined threshold
Replace p_1 and p_2 with their average value as the new bending point.

3 Attributed Graph Construction

By primitive extraction process, an engineering drawing has been decomposed into basic primitives, such as curves, straight lines, etc. To get a structural representation of the content of a engineering drawing, these primitives as well as their relationships must be well described. In this paper, the attributed graph is adopted to realize this purpose, where a graph node represents a primitive while the graph edges indicate the relationship between these primitives.

3.1 Graph Construction

Delaunay tessellation strategy is a natural way for graph construction from engineering drawing. In traditional Delaunay method, each primitive (curves, lines, etc.) is represented only by a point, such as middle point, as the input to graph construction. However, a long curve (line) in the original image may be near to several other primitives either at its middle part or at the end parts due to its long prolongation, therefore, a connection should be built from this primitive to each of these nearby primitives. If this long primitive is replaced just by its middle point, only the connections to these primitives which are near to this middle point will be survived, which leads to connections loss. Besides this, some false connections will also be generated due to the loss of primitive extension information.

To prevent the loss of primitive extension information, the following method is used in graph construction:

- (1) Sample a long primitive evenly into multiple points.
- (2) Adopt these sampled points as input to Delaunay tessellation.
- (3) Graph simplification. These graph nodes sampled from the same primitive are merged into one node, with their corresponding connections merged together.

By this sampling strategy, a primitive is represented by multiple points evenly distributed along this curve (line), therefore, the relationship with other primitives which are near to this primitive either at the middle part or at the end parts is reserved.

3.2 Attribute Extraction

Graph attributes play the key rule in characterizing the content of a engineering drawing, where node attributes depict the appearance of the primitives, such as circular, straight, or angular, while edge attributes define the spatial relationship between these primitives, such as parallel, intersectant, and so on.

In this paper, direction histogram is used to describe the appearance of each primitive. To realize rotation invariance, Fourier transform is carried out with this histogram and the coefficients of this transform are used as node attribute.

Directed edge attributes are proposed to highlight the relationship between two straight line segments. Let L_2 be the line under processed, and L_1 its neighbor line, the relationship from L_2 to L_1 is defined by following components:

- Relative angle α , say, the acute angle between L_2 and L_1 .
- Relative length rl : the length of L_2 divided by that of L_1 .
- Relative position rp , which describes the position of the intersection point.

If the intersection point of these two lines locates at L_2 , as shown in figure 2.1 and 2.2, this attribute is obtained through dividing the smaller length of line segment OD and OC by the larger one of them. Otherwise, it is calculated through dividing the length of OD by minus length of OC, as shown in figure 2.3.

- Relative distance rd . It is defined as the length of the line segment connecting the middle points of these two lines, divided by the length of L_2 .
- Relative minimum distance rmd . It is defined as the minimum distance between these two lines, divided by the length of L_2 .

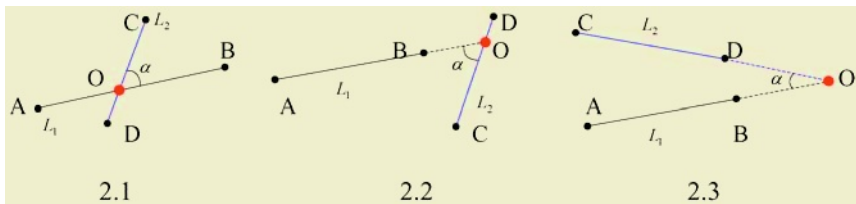


Fig. 2. Definition of relative position

These attributes, which are scale and rotation invariant, can precisely define the spatial relationship between two lines. However, our final target is to get the relative relation between two arbitrary curves. For this purpose, the curves are first approximated into straight lines, then, calculate the attributes between these two straight lines with above concept, which are used as the spatial relationship of original curves.

4 Graph Matching by Mean Field Theory

Graph matching is in fact a correspondence determination process [7, 8, 11], where the nodes in the query graph are seen as a set of labels to label these nodes in another

graph. Many algorithms have been proposed in the literature to solve this problem [7-11], such as mean field theory, relaxation labeling, genetic algorithm, etc. On the consideration of speed and performance, mean field theory is adopted in this paper for graph matching determination.

Given two graphs $g = (v_a, e_{a,b}), a, b = 1, \dots, A$ and $G = (V_i, E_{i,j}), i, j = 1, \dots, I$, where v_a and V_i represent the nodes of these two graphs, while $e_{a,b}$ and $E_{i,j}$ represent the directed edges of these two graphs, here, g and G are the abstraction graph corresponding to query drawing image and a database drawing image respectively.

Graph matching can be defined as follows: Find the match matrix M such that following cost function is minimized:

$$E(M) = \sum_{a=1}^A \sum_{i=1}^I \sum_{b=1}^A \sum_{j=1}^I M_{ai} M_{bj} Dis(e_{a,b}, E_{i,j}) + \sum_{a=1}^A \sum_{i=1}^I M_{ai} Dis'(v_a, V_i)$$

subject to $\forall a, \sum_{i=1}^I M_{ai} = 1; M_{ai} \in \{0,1\}$, where M_{ai} means node v_a in g is related to V_i in G .

The former part of this cost function represents the cost to match graph edges, the later part corresponds to the cost to match graph nodes together. $Dis(\cdot)$ represents the distance between two edges, while $Dis'(\cdot)$ the distance between two nodes.

With graph nodes and edges attributes described in section 3, the distance functions in above formula are defined as follows:

$$Dis(e_{a,b}, E_{i,j}) = \begin{cases} \text{cost} & \text{if } e_{a,b} \text{ or } E_{i,j} \text{ is NULL} \\ \frac{\alpha_1 - \alpha_2}{\pi i l_2} + \frac{|rp_1 - rp_2|}{2} + (1 - e^{-|rl_1 - rl_2|}) + (1 - e^{-|rd_1 - rd_2|}) + (1 - e^{-|rmd_1 - rmd_2|}) \end{cases}$$

where, $\{\alpha_1, rl_1, rp_1, rd_1, rmd_1\}, \{\alpha_2, rl_2, rp_2, rd_2, rmd_2\}$ are the attributes corresponding to these two edges, see section 3 for details. Above distance function is in fact a weighted average of attribute components, where the contribution of each element is normalized to one, and equal weights are adopted. It should be noted that the edges are directed due to the directed attributes defined in section 3.

$$Dis'(v_a, V_i) = \begin{cases} \text{cost}' & \text{if } v_a \text{ or } V_i \text{ is NULL} \\ \text{Euclidean distance of the Fourier Coefficients of these two nodes} \end{cases}$$

According to mean field theory, an iteration procedure is obtained to get the value of M :

$$q^{t ai} = -\frac{1}{T} \left(Dis'(v_a, V_i) + 2 \sum_{b=1}^A \sum_{j=1}^I M_{bj}^t Dis(e_{ab}, E_{i,j}) \right)$$

$$M_{ai}^{t+1} = \frac{\exp(q_{ai}^t)}{\sum_j \exp(q_{aj}^t)}$$

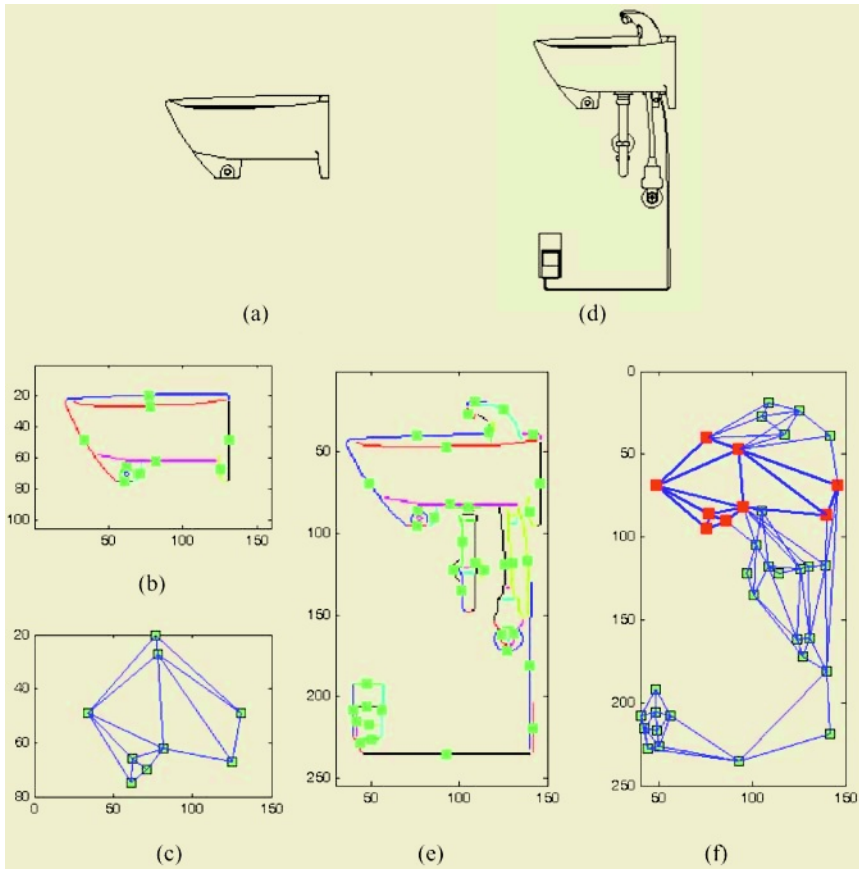


Fig. 3. Illustration of engineering drawings processing. (a) Query drawing; (b) Primitives extracted from (a); (c) Constructed graph of (a); (d) A database drawing; (e) Primitives extracted from (d); (f) Constructed graph of (d)

In matching, the temperature T is decreased towards 0 as iteration goes. The unambiguity of M is achieved when $T \rightarrow 0$. The convergence is also controlled by a quantity defined as:

$$Saturation = \frac{1}{A} \sum_a \sum_i M_{a,i}^2$$

also called the ‘saturation’ of M . The iteration is terminated if the saturation value is larger than a predefined value, such as 0.95.

5 Experiments

At the beginning of this section, an example is used to demonstrate the whole process described above, see figure 3. Figure 3(a) and 3(d) are original query drawing and database drawing to be matched; Figure 3(b) and 3(e) shows the extracted primitives from these two drawings, where a filled rectangle is used to denote a primitive. Figure

3(c) and 3(f) illustrate the graphs constructed. In figure 3(f) dark filled rectangles and thick lines are used respectively to denote the graph nodes and graph edges that are in correspondence with those of the query graph.

In this initial experiment, a database containing 100 engineering drawing images is set up to evaluate the retrieval performance of the proposed method. These images, which correspond to different designs of washbowls, are manually classified into three categories according to their appearances. The numbers of images in each category are 22, 22 and 56 respectively. The images in either first or second category are somewhat similar in that a common part can be found among them, however, the images in third category are distinct from each other. Figure 4 shows some representative images of each category.

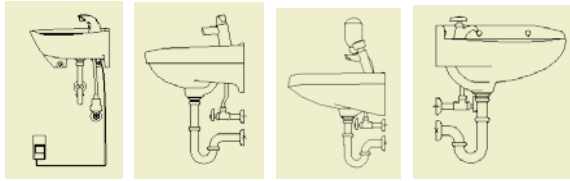


Fig. 4.1

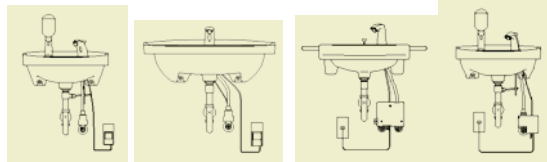


Fig. 4.2

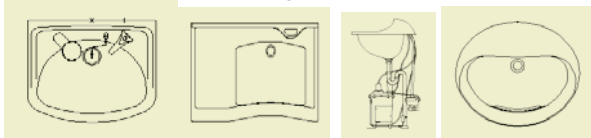


Fig. 4.3

Fig. 4. Example drawings of the database (Fig. 4.1 First category, Fig. 4.2 Second category, Fig. 4.3 Third category)

In the experiments, we generate totally 20 query images to evaluate the retrieval performance, with each 10 of them being similar to the common part of first and second category respectively, and the average precision and recall values are obtained from these trails. Figure 5 shows an example of retrieval result, where the former 20 images are displayed. In figure 5.2, rectangles are used to denote those parts in the database drawings that are similar to query one. The average precision and recall value are illustrated in figure 6.

In former 22 retrieved images, which is the true number of each category, the average precision is about 56%, and the recall value is more than 80% in the former 50 retrieved images. This retrieval can largely facilitate the designers in searching a specific previous drawing, thus saves the cost in designing a new product. All these experiments are carried out at a P4-1.3G computer, and the average time used for once retrieval is 111.5 seconds, that is to say, the average time for matching one database engineering drawing with the query one is about 1.1 seconds.

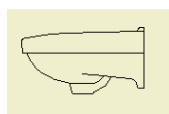


Fig. 5.1

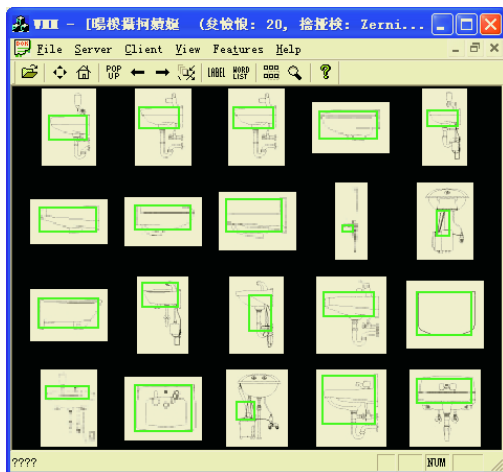


Fig. 5.2

Fig. 5. An example of retrieval result (Fig. 5.1 Query image, Fig. 5.2 Retrieval result, with former 20 retrieved images shown)

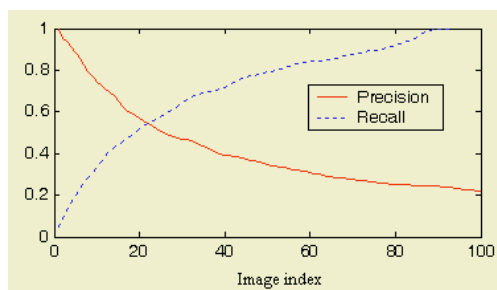


Fig. 6. Retrieval performance

6 Conclusions

In this paper, we present a practical graph matching based method for engineering drawings retrieval. By structural representation, the content as well as the spatial relationship of the engineering drawing are well described. While the reasonable primitives in an engineering drawing, say curves, straight lines, ellipses and so on, can be extracted by the merge-split process proposed in this paper. Thus, engineering drawings retrieval becomes a problem of comparing the structural representations, which can be implemented by graph matching.

In this paper, it is assumed that all lines and curves in the object image are one-pixel width in clean background. However, as a kind of method, the technique described in this paper is not limited to such images only. By some pre-processing, such as de-noise, bar detection, etc, the rough primitives can be obtained from any kinds of engineering drawings. After this, above method can be used to form reasonable primitives, build structural descriptions, and matching for retrieval.

References

1. K. Tombre, Analysis of Engineering Drawings: State of the Art and Challenges, in K. Tombre, A.K.Chhabra, Graphics Recognition – Algorithms and Systems, 257-264
2. T. Kanungo, R. M. Haralick, D. Dori, Understanding Engineering Drawings: A Survey, International Workshop on Graphics Recognition, pp. 119-130, 1995
3. R. Kasturi, S. T. Bow, W. El-Masri, et al, A System for Interpretation of Line Drawings, IEEE Trans. On P.A.M.I., Vol. 12, No. 10, pp. 978-992, 1990
4. Dov Dori, Wenyin Liu, Sparse Pixel Vectorization: an algorithm and its Performance Evaluation, IEEE Trans. on P.A.M.I., Vol. 21, No. 3, pp. 202-215, 1999
5. Jiqiang Song, Feng Su, et al, An Object-Oriented Progressive-Simplification-Based Vectorization System for Engineering Drawings: Model, Algorithm, and Performance, IEEE Trans. on P.A.M.I., Vol. 24, No. 8, pp. 1048-1060, 2002
6. Benoit Huet, E. R. Hancock, Relational Object Recognition from Large Structural Libraries, Pattern Recognition, Vol. 35, pp. 1895-1915, 2002
7. W. J. Christmas, J. Kittler, M. Petrou, Structural Matching in Computation Vision using Probabilistic Relaxation, IEEE Trans. on P. A. M. I., Vol. 17, No. 8, pp. 749-764, 1995
8. Steven Gold, Anand Rangarajan, A Graduated Assignment Algorithm for Graph Matching, IEEE Trans. on P. A. M. I., Vol. 18, pp. 377-388, 1996
9. S. Z. Li, H. Wang, K. L. Chan, Energy Minimization and Relaxation Labeling, Journal of Mathematical Imaging and Vision, Vol. 7, pp. 149-161, 1997
10. M. A. Wyk, T. S. Durrani, B. J. Wyk, A RKHS Interpolator-Based Graph Matching Algorithm, IEEE Trans. on P. A. M. I., Vol. 24, No. 7, pp. 988-995, 2002
11. A. D. J. Cross, R. C. Wilson, E. R. Hancock, Inexact Graph Matching using Genetic Search, Pattern Recognition, Vol. 30, No. 6, pp. 953-970, 1997