Spotting Symbols in Line Drawing Images Using Graph Representations

Rashid Jalal Qureshi, Jean-Yves Ramel, Didier Barret, and Hubert Cardot

Université François-Rabelais de Tours
Laboratoire d'Informatique (EA 2101)
64, Avenue Jean Portalis, 37200 Tours – France
{rashid.qureshi,jean-yves.ramel,
didier.barret,hubert.cardot}@univ-tours.fr

Abstract. Many methods of graphics recognition have been developed throughout the years for the recognition of pre-segmented graphics symbols but very few techniques achieved the objective of symbol spotting and recognition together in a generic case. To go one step forward through this objective, this paper presents an original solution for symbol spotting using a graph representation of graphical documents. The proposed strategy has two main step. In the first step, a graph based representation of a document image is generated that includes selection of description primitives (nodes of the graph) and organisation of these features (edges). In the second step the graph is used to spot interesting parts of the image that potentially correspond to symbols. The sub-graphs associated to selected zones are then submitted to a graph matching algorithm in order to take the final decision and to recognize the class of the symbol. The experimental results obtained on different types of documents demonstrates that the system can handle different types of images without any modification.

Keywords: Document Image Analysis, Graphics Symbols Recognition, Symbol Spotting, Raster-to-Vector Techniques, Line drawings.

1 Introduction

The classical approaches of symbol recognition use knowledge about the document to drive the analysis of the image [1] [2]. The segmentation algorithms are often looking directly in the bitmap images for precise information specified in the model databases. So, the image analysis is achieved by using mostly a split strategy forced by a priori knowledge. Two classical processing ways can be mentioned:

- Methods looking for some particular configurations of pixels or of low level primitives in the image (horizontal – vertical lines, specific textures, closed loops, connected components …) corresponding to potential symbols [3].
- Methods based on computation of signatures for identification and recognition of symbols. Recently presented in [4] [5], signatures are a collection of features extracted from pre-segmented shapes. Signatures of candidate symbols are computed and matched against the existing signatures of all the models stored in the library of a particular document. In most of the cases, the "localisation" is

W. Liu, J. Lladós, and J.-M. Ogier (Eds.): GREC 2007, LNCS 5046, pp. 91–103, 2008.

realised using the connected components or buckets (small square part of the image). In this last case, the test images are divided into buckets and signature of each bucket is computed and is matched against the symbols signatures to determine what kind of symbols a bucket is likely to contain. With such methods, as pointed out by P. Dosch [4], a lot of false alarms are still present, especially with parts of the images not presented in the pre-definite library. Furthermore, the recognition rate may decrease on much larger databases and the strength of classification may suffer in case of noisy images.

An interactive approach to recognition of graphic objects in engineering drawings has been recently proposed by L. Wenyin [6]. Interactively, the user provides examples of one type of graphic object by selecting them in an engineering drawing and then the system learns the graphical knowledge and uses this learnt knowledge to recognize or search for other similar graphic objects in the same image. However, it is desired that we could segment graphic symbols automatically in large dataset without human involvement and user feedback for each analysed image. A weak aspect which is common in all these algorithms is the lack of information about the handling of regions which do not belong to one of the expected categories.

2 Graph Representation of Image Content

The proposed approach relies on structural methods and is based on capturing the spatial and topological relationships between graphical primitives. The pre-processing includes vectorization of the raster image. The technique which we had proposed in [7] is used to constructs a quadrilaterals based representation of lines in a drawing (figure 1, figure 2b). Each quadrilateral has attributes like length (ℓ) of the median axis, angles of the two vectors (v_1, v_2), width on each side (w_1, w_2) and a zone of influence (as shown by dotted rectangle in figure 2c). The dimensions of the zone of influence depends on the length (ℓ) and the two widths (w_1, w_2) of the quadrilateral i.e., $U_x = \ell / 4$ and $U_y = ((w_1 + w_2) / 2) \times 4$. In fact, in our structural representation, nodes represent quadrilaterals and edges represent the topological relationships between neighbouring quadrilaterals (figure 2d). The length of the quadrilateral is associated to the nodes as an attribute, and relative angle i.e., angle between two neighbouring quadrilaterals is associated to edges as an attribute. All edges are also associated with one label representing the type of the topological relationship (L-junction, S-junction, T-junction, X-intersection or P-parallelism) that exists between the two neighbouring quadrilaterals (figure 2e, figure 2f). A detailed description of this graph representation can be found in [7] [8].

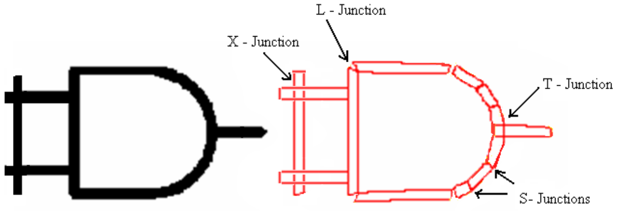


Fig. 1. Initial image (on left), Vectorization results (on right)

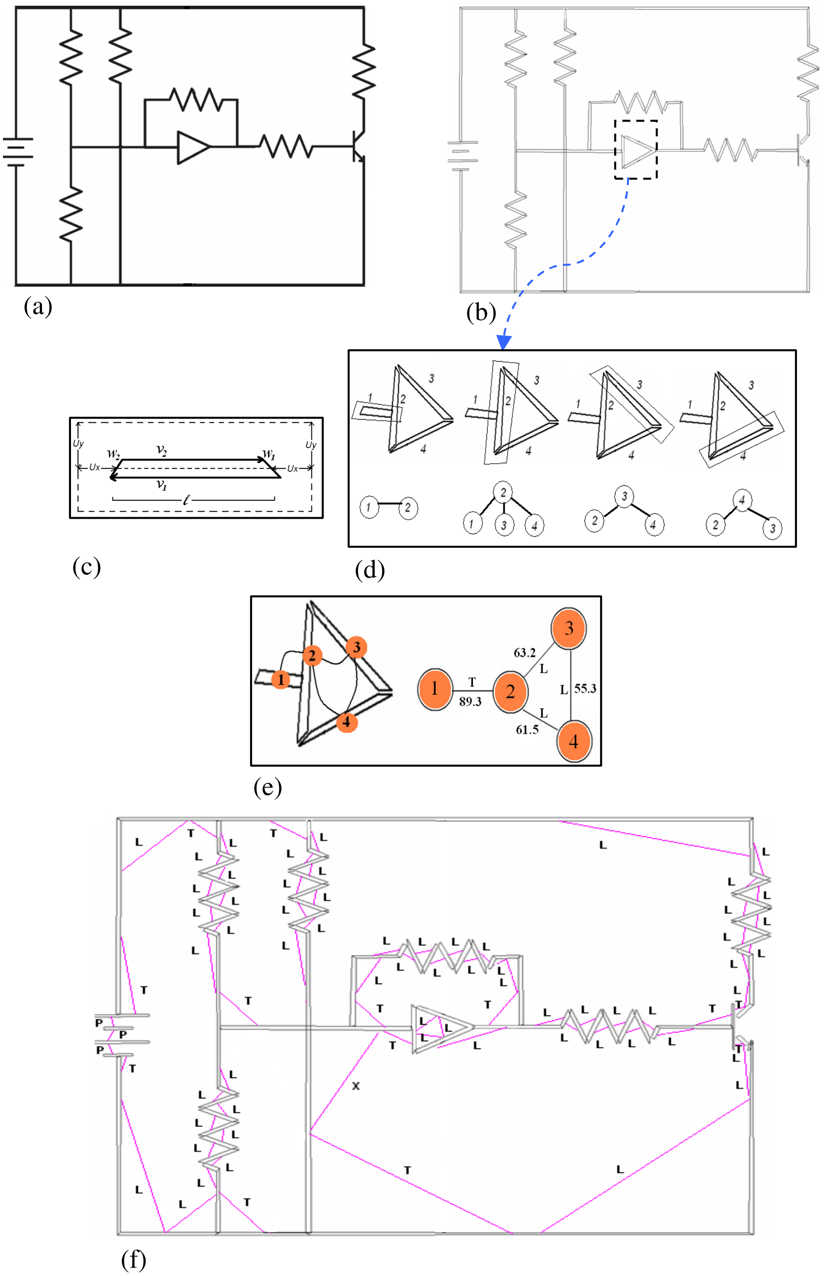


Fig. 2. (a) Initial image, (b) Vectorization results, (c) Zone of influence of a quadrilateral (d) Influence zone of the quadrilaterals and their corresponding sub-graphs respectively, (e) and (f) Graph representation

3 Seeds Detection in the Graph

The key idea of our method is to detect the parts of the graph that may correspond to symbol without a priori knowledge about the type of the document. Such nodes and edges will constitute symbol seeds. Then, the seeds will be analyzed and grouped together to generate sub-graphs that are potentially corresponding to symbols in the document image. We observed from structure analysis of drawings, circuits, maps and diagrams that symbols are composed of quadrilaterals with specific characteristics easily detectable in the graph representation. So, scores (probabilities of being part of a symbol) can sequentially be associated to all the edges and all the nodes of the graph to provide the symbol seeds. The hypothesis used to compute the scores of the nodes and the edges are:

H_1 - Symbols are composed of small segments compared to other elements of the drawings

H_2 - The segments constituting a symbol have comparable lengths

H_3 - Two successive segments with a relative angle far from 90° have a higher probability of being part of a symbol

H_4 - Symbols are often composed of parallel segments

H_5 - A symbol segment is rarely connected to more than 3 other segments

H_6 - Shortest loops are most often corresponding to symbols

This is the basic set of hypothesis that our system uses for the purpose of symbols spotting in documents images. However, the documents containing symbols that do not respect these hypotheses can not be analyzed using the proposed system.

3.1 Studies of the Nodes and the Edges

A study of graph nodes and edges characteristics and attributes is made and a score between 0 and 1 is computed for each node and edge using the above hypothesis. The score of an edge is a function of the two relative lengths and angles of the connected nodes (quadrilaterals) and of the type of topological relationship between them (eq.1).

The score of a node (eq. 2) is computed by using the accumulated scores of its connecting edges, the number of its connecting edges and the length of the corresponding quadrilateral.

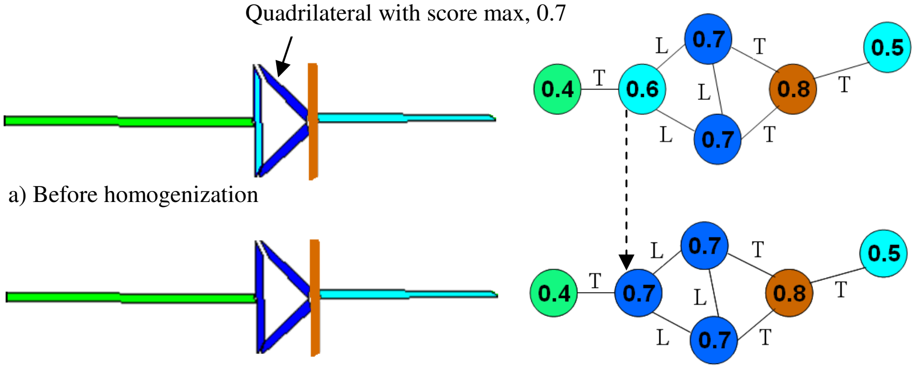
$$Score(E_i) = \alpha.P_{E1}(E_i \rightarrow Type)+\beta.P_{E2}(E_i \rightarrow Angle) + \gamma.P_{E3}(E_i \rightarrow N_1 \rightarrow Length,\ E_i \rightarrow N_2 \rightarrow Length) \quad (1)$$

$$Score(N_i) = \lambda\ \left(\frac{\sum_{j=1}^{\Omega(N_i)} Score(Ej)}{\Omega(Ni)} \right) + \mu\ .\ P_{N2}(\Omega(N_i)) + \omega\ .P_{N3}(N_i \rightarrow Length) \quad (2)$$

In eq.1, E_i is an edge of the graph, α, β and γ are weights and P_{Ei} are functions returning scores between 0 and 1 according to the %age agreement with the above mentioned hypothesis. Here α, β and γ are set to *1/3* corresponding to the weights

Table 1.

Functions	Scores
$P_{E1}(E_i \rightarrow Type)$	0.50 (T), 0.25 (L), 0.50 (X), 1.0 (P), 0.50 (S)
$P_{E2}(E_i \rightarrow Angle)$	$\lvert 90-(E_i \rightarrow Angle)\rvert / 90$
$P_{E3}(E_i \rightarrow N_1 \rightarrow Length,\ E_i \rightarrow N_2 \rightarrow Length)$	$\frac{\min(N_1 \rightarrow Length, N_2 \rightarrow Length)}{\max(N_1 \rightarrow Length, N_2 \rightarrow Length)}$
$P_{N2}\,(\Omega(N_i))$ *(Ω : degree of a node)*	0.25 ($\Omega < 2$) 1.00 ($2 \leq \Omega \leq 4$) 0.25 ($\Omega > 4$)
$P_{N3}\,(N_i \rightarrow Length)$ (Length of the primitive in pixels)	0.25 ($Length < 20$) 1.00 ($20 \leq Length \leq 200$) 50/*Length* ($Length > 200$)



a) Before homogenization

b) After homogenization, 0.7 allocated to all the quadrilaterals connected in loop

Fig. 3. Propagation of the maximum score to all the nodes in the path

associated to each hypothesis. In eq.2, N_i is a node of the graph, $\Omega(Ni)$ is the degree of the node Ni, ω, μ and λ are weights and P_{Ni} are functions returning scores between 0 and 1 according to $H_1 - H_6$, $\lambda = 1/2$ and $\mu = \omega = ¼$ corresponding to the weights associated to each hypothesis.

The values computed by functions P depends on the attributes associated with nodes and edges of the graph, as shown in Table.1. All these weights have been fixed in the light of statistics collected from different images of the databases (electrical diagrams, logic diagrams, architectural maps).

3.2 Score Propagation in the Graph

The different seeds are associated and extended by merging with other seeds depending on their relative scores and relation. This score propagation process seeks and analyses the different loops with shortest length between seeds (nodes in the graph). This step only modifies the scores of the nodes modeling quadrilaterals that form closed loop. The scores of all the nodes belonging to a detected path are homogenized (propagation of the maximum score to all the nodes in the path) until convergence. This is the phase where edges with L and S attributes act as bridges between the different parts of a spotted symbol.

This step only modifies the scores of the symbols composed of closed loops but has no influence on other parts of the graph.

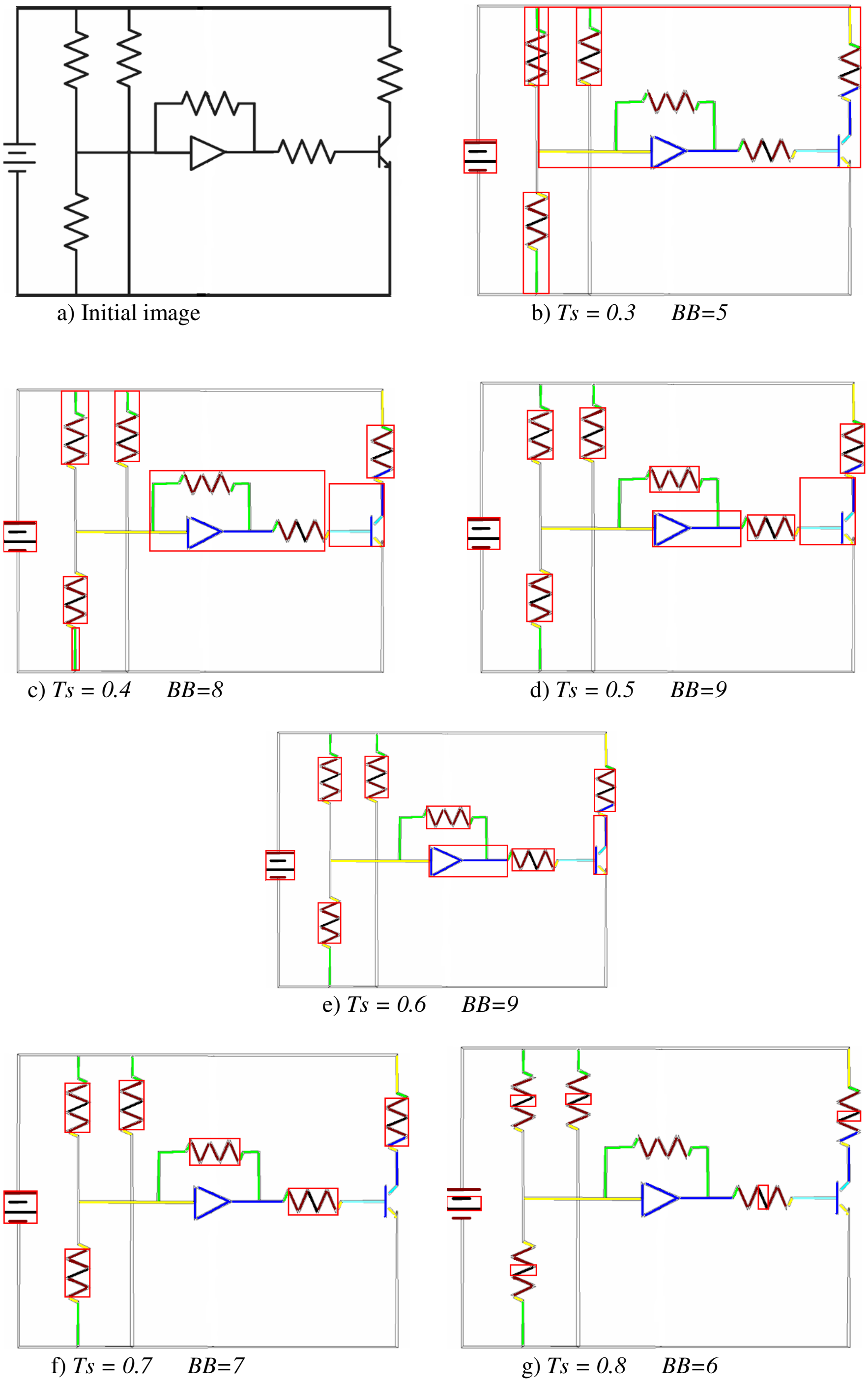
4 From Seeds to Symbol Spotting and Recognition

4.1 Generation of the Bounding Boxes

After the computation of the scores of the nodes, a sub-graph extraction has to be achieved in order to provide the symbol recognizer module with potential symbol present in the document. To obtain these sub-graphs corresponding to symbols and the associated bounding boxes (figure 5), our methods looks for all the sets of directly

```
Algorithm-1
Input : Graph representation of a graphic document
Output : Number of bounding boxes(spotted symbols), Best value of seed threshold $Ts$
                    List of Bounding boxes$(BB)$, List of sub-graphs$(SG)$

for $Ts = 0$   to 1 with step = 0.1 do
  begin
     $Max\_BB \leftarrow 0$        // Maximum number of bounding boxes
     $Best\_Ts \leftarrow 0$        // Best threshold value

      for $j = 1$ to $| E |$ do
         compute: score($E_j$)

      for $i = 1$ to $| N |$ do
         compute: score($N_i$)

         Seeds $\leftarrow$   $\forall (N_i) \mid \text{score}(N_i) \geq Ts$
         Propagation of scores of quads in loops with $L$ and/or $S$ type
connections

         $SG \leftarrow$ Get Sub-graphs
         $BB \leftarrow$ Get Bounding Boxes

       if ($BB > Max\_BB$)
           $Max\_BB \leftarrow BB$
           $Best\_Ts \leftarrow Ts$
       end if
      end for
```

Fig. 4. Selection of seed threshold (*Ts*) automatically



a) Initial image

b) *Ts = 0.3* *BB=5*

c) *Ts = 0.4* *BB=8*

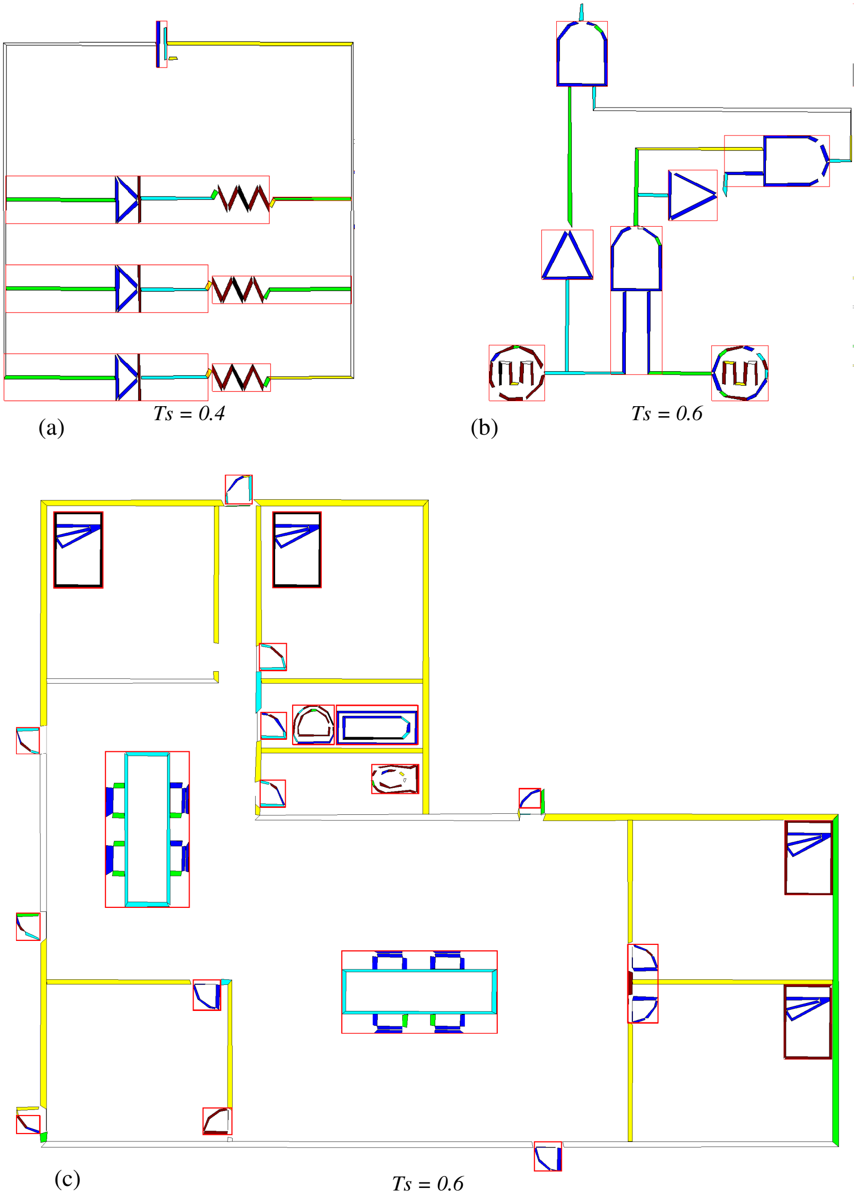
d) *Ts = 0.5* *BB=9*

e) *Ts = 0.6* *BB=9*

f) *Ts = 0.7* *BB=7*

g) *Ts = 0.8* *BB=6*

Fig. 5. Influence of seed threshold (*Ts*) on symbols spotted (Bounding Box)

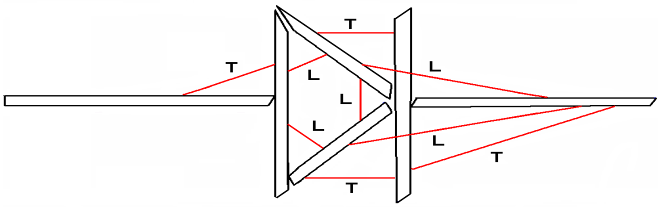


⬛ = scores from 1 to 0.7 🟦 = scores from 0.7 to 0.6 🟦 = scores from 0.6 to 0.5

🟩 = scores from 0.5 to 0.4 🟨 = scores from 0.4 to 0.3 - ⬜ = scores from 0.3 to 0

if threshold *Ts* = 0.6, only ⬛ 🟦 are considered as seeds

Fig. 6. Seeds and spotted sub-graphs



```
<Graph id="Symbole4">
  <node id="node0">
    <attr name="forme"><string>Quad</string></attr>
    <attr name="x1i"><string>232</string></attr>
    <attr name="y1i"><string>497</string></attr>
    <attr name="x1f"><string>231</string></attr>
    <attr name="y1f"><string>417</string></attr>
    <attr name="x2i"><string>227</string></attr>
    <attr name="y2i"><string>418</string></attr>
    <attr name="x2f"><string>229</string></attr>
    <attr name="y2f"><string>498</string></attr>
    <attr name="angle1"><string>90</string></attr>
    <attr name="angle2"><string>91</string></attr>
    <attr name="thickness1"><string>8</string></attr>
    <attr name="thickness2"><string>7</string></attr>
    <attr name="length"><string>81</string></attr>
    <attr name="score"><string>0.703</string></attr>
  </node>
    …
    …
    …
<edge id="edge1" from="node0" to="node1">
    <attr name="angle"><string>89</string></attr>
    <attr name="type"><string>T</string></attr>
    <attr name="score"><string>0.409</string></attr>
</edge>
<edge id="edge2" from="node1" to="node2">
    <attr name="angle"><string>46</string></attr>
    <attr name="type"><string>L</string></attr>
    <attr name="score"><string>0.388</string></attr>
</edge>
…
…
…
</graph>
</gxl>
```

Fig. 7. Example of sub-graph encoded with GXL and extracted from Fig. 3a

connected seeds (nodes) in the graph using a recursive process. Only, nodes having highest score (probability of being part of symbols) have to be considered as part of the symbols (figure 5).

It is possible to vary the seed threshold (*Ts*) and to compare the number of symbols generated manually as well as automatically (figure 4). A simple rule for automatic selection of seed threshold (*Ts*) is to consider the value which gives maximum number of bounding box (figure 5). The experiments demonstrated that the better results are obtained by keeping the threshold providing the maximum number of symbols in the image.

However, we can test other complex rules by using statistics about these bounding boxes and ultimately the seeds inside, for example, the dimensions of the bounding boxes.

4.2 Symbol Recognition or Rejection

Hence, we used our graph matching routine [8] to match these sub-graphs with graph representation of models. The detected zones i.e., sub graphs are matched against model graphs using polynomial bound greedy algorithm for the symbol recognition task.

This recognition algorithm outputs a score of similarity and the best mapping of nodes found. The graph matching is error-tolerant and works well in case of under or over segmentation of symbols. The score of similarity produced by the matching algorithm can easily be used (using a threshold) to automatically decide if the extracted sub-graph corresponds to a symbol or not (rejection of the zone).

The approach is parallel, and is capable of spotting all the symbols present in a drawing in one pass. The different steps of proposed strategy are shown in figure 8, however in this paper we have detailed the steps related to symbol spotting only.

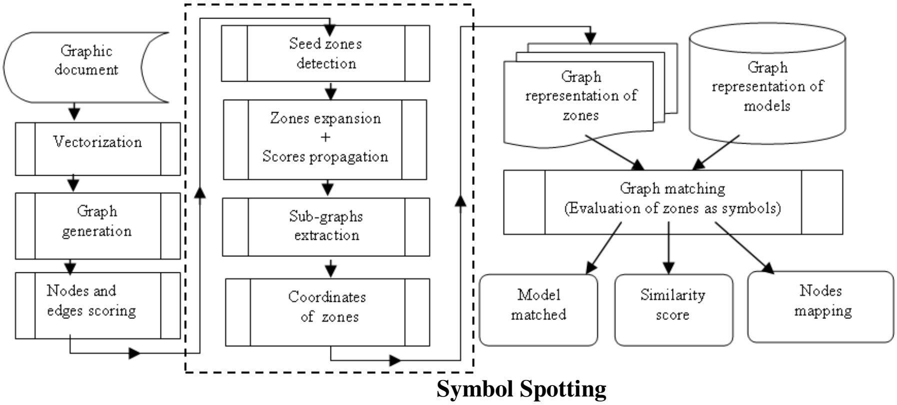
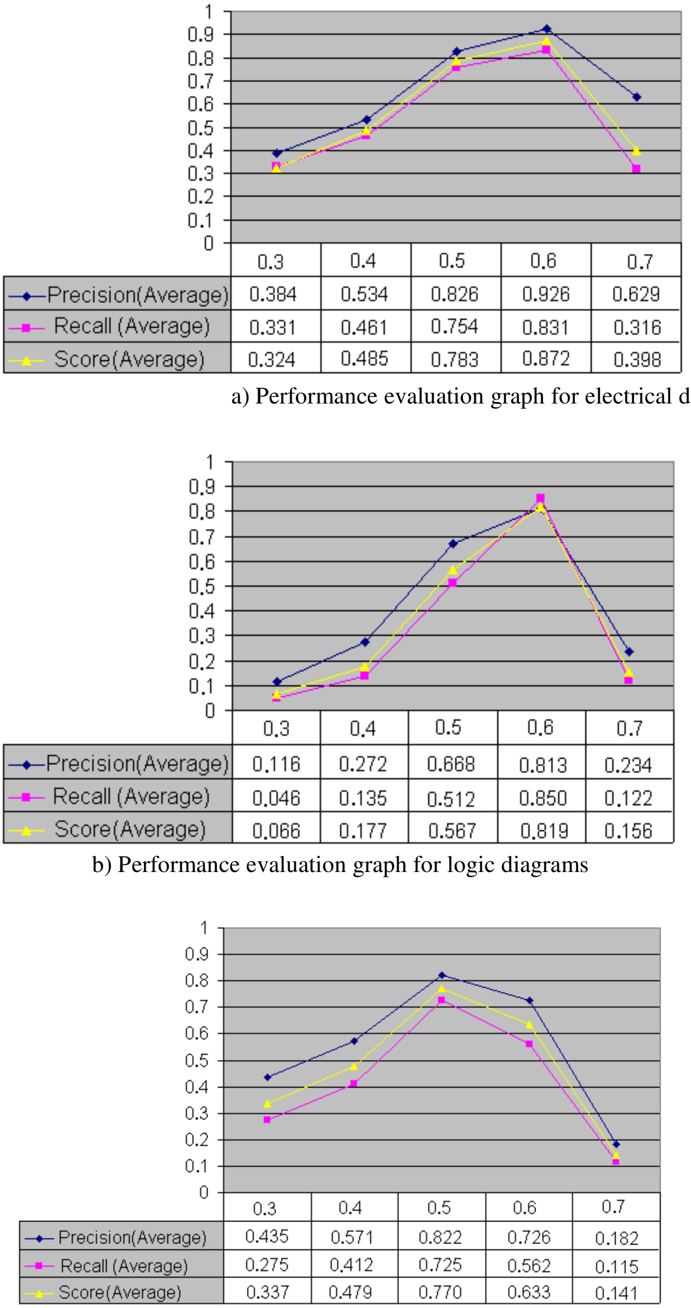


Fig. 8. Proposed system architecture

5 Results and Conclusion

Tests have been conducted on three types of graphic documents, electronic circuits, logic diagrams and architectural maps.

To evaluate the performance of the proposed system we have followed the general framework based on the notion of precision and recall presented in [9]. For a given test, let *T* be the number of targets belonging to the ground-truth, and *R* the set of



	0.3	0.4	0.5	0.6	0.7
◆ Precision(Average)	0.384	0.534	0.826	0.926	0.629
■ Recall (Average)	0.331	0.461	0.754	0.831	0.316
▲ Score(Average)	0.324	0.485	0.783	0.872	0.398

a) Performance evaluation graph for electrical diagrams

	0.3	0.4	0.5	0.6	0.7
◆ Precision(Average)	0.116	0.272	0.668	0.813	0.234
■ Recall (Average)	0.046	0.135	0.512	0.850	0.122
▲ Score(Average)	0.066	0.177	0.567	0.819	0.156

b) Performance evaluation graph for logic diagrams

	0.3	0.4	0.5	0.6	0.7
◆ Precision(Average)	0.435	0.571	0.822	0.726	0.182
■ Recall (Average)	0.275	0.412	0.725	0.562	0.115
▲ Score(Average)	0.337	0.479	0.770	0.633	0.141

c) Performance evaluation graph for architectural maps

Fig. 9. Precision / Recall graphs for few prototypes of electrical diagrams, logical diagrams and architectural maps

results supplied by an application. The number of exact results is called *e*. The precision is then defined as the number of exact results divided by the number of results:

$$p = e / | R |$$

The recall *r* is defined as the number of exact results divided by the number of targets:

$$R = e / | T |$$

The precision and recall then combined to determine the global score *s*, expressing the recognition rate:

$$s = \frac{2}{(1/p) + (1/r)}$$

The localization rate (without considering the recognition step) depends on the threshold associated to seed scores (figure 9).

Localization rates are better in electronic circuits and in logic diagrams where symbols are quite clear. However, the segmentation rate decreases in case of architectural maps where symbols are connected with lines representing walls. In this case, it is preferable to choose a low threshold for seed selection not to miss too many symbol seeds. The recognition step can be used to validate the spotted bounding boxes in connection with proposed rejection mechanism.

We conclude with the remarks that, developing a flexible and general framework for symbol localization and recognition is really a challenge due to large variation in the basic elements of graphic documents. The proposed approach has shown good results on the three different types of graphic documents tested so far (without complex/difficult modification of the parameters used by the system). The method is parallel, and is capable of spotting the symbols present in a drawing in one pass. The graph matching is error-tolerant and works well in case of under or over segmentation of symbols.

References

1. Joseph, S.H., Pridmore, T.P.: Knowledge Directed Interpretation of Mechanical Engineering Drawings. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(9), 928–940 (1992)
2. DenHartog, J.E., TenKate, T.K., Gerbrands, J.J.: Knowledge Based Interpretation of Utility Maps. Computer Vision and Image Understanding 63(1), 105–117 (1996)
3. Song, J., Su, F., Tai, M., Cai, S.: An Object-Oriented Progressive-Simplification-Based Vectorization System for Engineering Drawings: Model, Algorithm, and Performance. IEEE Transaction on Pattern Analysis and Machine Intelligence 24(8), 1048–1060 (2002)
4. Lladós, J., Dosch, P.: Vectorial Signatures for Symbol Discrimination. In: Lladós, J., Kwon, Y.-B. (eds.) GREC 2003. LNCS, vol. 3088, pp. 154–165. Springer, Heidelberg (2004)
5. Wang, Y., Phillips, I.T., Haralick, R.M.: Document Zone Content Classification and its Performance Evaluation. Pattern Recognition 39, 57–73 (2006)

6. Wenyin, L., Zhang, W., Yan, L.: An Interactive Example-Driven Approach to Graphics Recognition in Engineering Drawings. International Journal on Document Analysis and Recognition 9, 13–29 (2007)
7. Ramel, J.Y., Vincent, N., Emptoz, H.: A Structural Representation for Understanding Line Drawing Images. International Journal on Document Analysis and Recognition 3(2), 58–66 (2000)
8. Qureshi, R.J., Ramel, J.Y., Cardot, H.: Graphic Symbol Recognition Using Flexible Matching Of Attributed Relational Graphs. In: Proceeding of 6th IASTED International Conference on Visualization, Imaging, and Image Processing (VIIP), Palma de Mallorca-Spain, pp. 383–388 (2006)
9. Valveny, E., et al.: A general framework for the evaluation of symbol recognition methods. IJDAR 9, 59–74 (2007)