Electronic Letters on Computer Vision and Image Analysis 5(2):47-57, 2005

Frequent Graph Discovery: Application to Line Drawing Document Images

Eugen Barbu, Pierre Héroux, Sébastien Adam, and Éric Trupin

Laboratoire PSI
CNRS FRE 2645 - Université de Rouen
76 821 Mont-Saint-Aignan cedex - France

Received 16 July 2004; accepted 16 November 2004

Abstract

In this paper a sequence of steps is applied to a graph representation of line drawings using concepts from data mining. This process finds frequent subgraphs and then association rules between these subgraphs.

The distant aim is the automatic discovery of symbols and their relations, which are parts of the document model. The main outcome of our work is firstly an algorithm that finds frequent subgraphs in a single graph setting and secondly a modality to find rules and meta-rules between the discovered subgraphs. The searched structures are closed [1] and disjunct subgraphs. One aim of this study is to use the discovered symbols for classification and indexation of document images when a supervised approach is not at hand. The relations found between symbols can be used in segmentation of noisy and occluded document images. The results show that this approach is suitable for patterns, symbols or relation discovery.

Key Words: Computer Vision, Image Analysis, Pattern Recognition, Graph Mining, Line Drawings, Association Rules.

1 Introduction

A symbol encodes a message into the form of an arbitrary sign. This sign has acquired a conventional significance. According to the document model, the symbol conveys graphical and semantic information. In this paper we try to discover both the representation as a written sign, and the relations (rules) that a symbol respects. The graphical representation and the rules found can be considered as an approximation of the message carried by the symbol. Automatic symbol extraction on document images without any prior domain knowledge is an appealing task. This approach has been pursued by Altamura [2] and Messmer [3]. In the context of line drawings document, one way to detect symbols is to consider the frequent occurrences of included entities. The entities can be graphs, geometric shapes or image parts depending at which processing level (segmentation) we apply this method [4], [5], [6]. A possible extension of this approach is to find relations between symbols. Such a relation can be viewed as a new entity that can be frequent and participates on its own right in other more complex relations. The standard for mining frequent item sets is the A priori algorithm [7]. However if the objects are graphs, some modifications to the basic algorithm

Correspondence to: eugen.barbu@univ-rouen.fr

Recommended for acceptance by J.M. Ogier, T. Paquet, G. Sanchez
ELCVIA ISSN: 1577-5097
Published by Computer Vision Center / Universitat Autonoma de Barcelona, Barcelona, Spain

should be made. Several papers describe A priori-like algorithms for mining frequent graph substructures [8], [9], [10].

This paper presents an algorithm that finds frequent subgraphs in a graph, a modality of creating rules and meta-rules between the discovered symbols and some possible utilization for the detected rules.

The principle of our approach is described on Fig.1.

A document image is characterised in a certain extent by the set of symbols that are frequent. Using this incomplete description of a document, generated in an unsupervised manner, we can use techniques from Information Retrieval in order to index [11] and classify [12] document images.

A good example for using the rules between objects can be to cluster a set of document images. If the symbols are described in the common graph language, the rules can also be shared. Two documents are from the same class if they respect the same rules. The distance between two documents can be evaluated using the extent to which one document conforms to the rules of the other.

Another application of the rules between symbols is to apply these rules in the segmentation process when noise or occluded symbols are present.

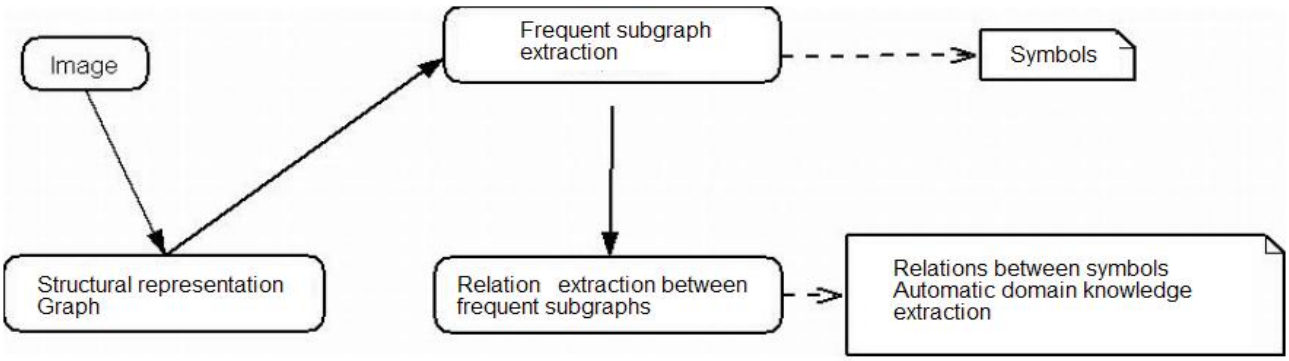


Fig. 1. Approach principle

This paper is organized as follows. Section 2 addresses the algorithm for finding frequent subgraphs. Section 3 emphasizes the ways we can find association rules between symbols. Section 4 presents an example of the proposed method. Section 5 elaborates several conclusions.

2 An algorithm that finds frequent subgraphs

The proposed approach is based on the fact that symbols on technical drawings graphically encode message elements according to a certain convention. So, in several document images sharing the same document model, a pattern always describes the same entity. The symbols of a document class appear with a certain frequency.

The purpose of this algorithm is to find the frequent subgraphs from a graph that describes the neighbourhood relations between shapes in a line drawing document. The subgraphs which represent symbols are closed graphs (a graph is closed if it does not have a super-graph with the same number of apparitions in the dataset) [1].

In the process of document image analysis, different graph based representations can be used. These representations can be constructed depending on the understanding level of the document when the graph is generated or according to the type of document that one tries to model (mostly textual, mostly graphical, mixed…)

In this paper we extract a graph from the document image at a low level of document understanding. We only use connected components and their neighbouring relations to construct the graph. The documents analysed are mostly graphical documents called line drawings. From a semantic point of view, a line drawing document is a document that does not lose information when the morphological operation of skeletonisation is applied on it.

The document graph is obtained from a line drawing considering:

- the regions (closed loops, two-dimensional shapes) or one-dimensional shapes as nodes.
- the neighbouring relations between these shapes as edges.

Two shapes are neighbours if they share a common frontier (see Fig. 2). This relation of neighbourhood can also be computed using a distance between node regions. One example can be: two occlusions are neighbours if the distance between their centers is less than a fixed or relative threshold. This representation is more robust than the binary relation of neighbourhood computed using the existence or not of a common frontier but has the disadvantage of using a more or less arbitrary threshold.

In order to label each node we extract a vector of features called Zernike moments for every part of the image that represents a node of the representation graph. These features are rotation invariant. More properties on these features can be found in [13].

We apply an unsupervised clustering algorithm on the nodes of the representation and each node has the class it belongs to as label. The clustering algorithm used is hierarchical ascendant, clustering using the Euclidean distance as dissimilarity, complete-linkage distance between clusters, and the Calinsky-Harabasz index to obtain the number of clusters. This algorithm has been chosen after a comparison with a hierarchical descendant clustering using the Duda-Hart index as stopping criterion and based on the conclusions from [14].

Two graphs represent the same symbol if they are isomorphic and if each pair of nodes (associated by the isomorphism function) has the same label.

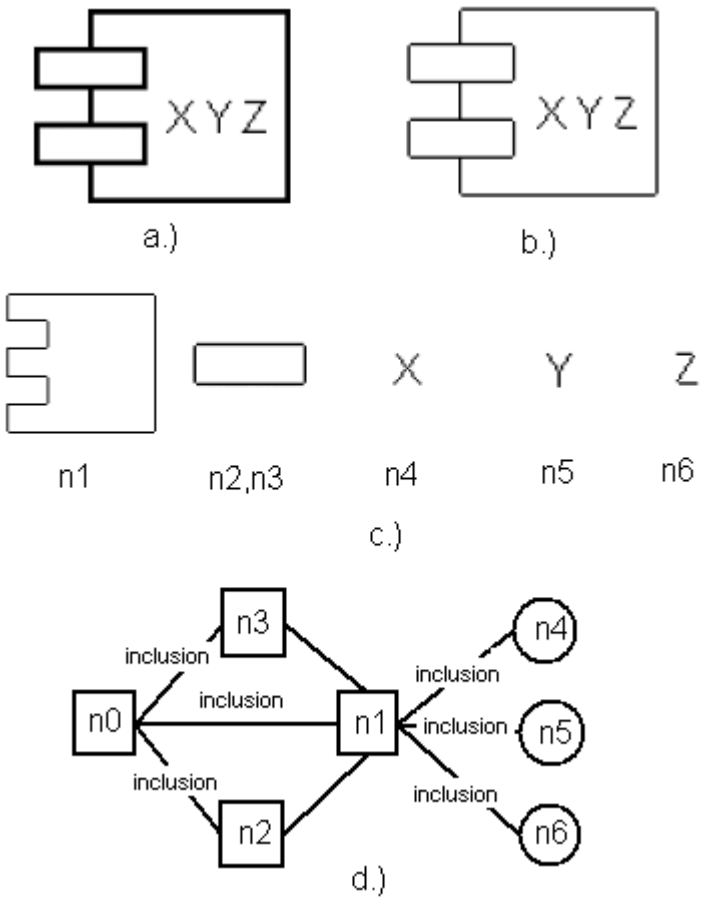


Fig. 2. A drawing a.) and its associated graph d.), considering the background region *n0*. The *1-dimensional* shapes are represented by circles. The *2-dimensional* shapes are represented by rectangles.

In this context a subgraph is considered frequent if its number of apparitions as non-included in other subgraphs is greater than a certain threshold *s*.

The way the threshold is defined can be linked to two possible settings: single or multiple graphs. In multiple graphs setting, i.e. we have a set of graphs and each graph is called a "transaction", we can say a subgraph is frequent if it appears in more than $^2s\%$ transactions. In our case we are interested in the frequent occurrences of a subgraph in the same graph, so we are in a single graph setting.

Because the number of subgraphs of the same class (any two subgraphs from the same class are isomorphic) is considered for a single graph, the threshold cannot be defined in relation with the number of transactions as it is done in other similar algorithms ([9], [10]). Considering a single transaction, we are interested in symbol occurrences included in that transaction. Here the threshold s is computed considering an approximation of the maximum possible number of subgraphs, with disjoint node sets and fixed number of edges and nodes, contained in the document graph.

The proposed algorithm uses the principle behind "A priori"-like algorithms combined with two simplifying hypotheses:

- the symbols are rarely expressed by graphs with a large number of nodes (10)
- occurrences for the same symbol are subgraphs with disjoint node sets

The idea behind all A priori-like algorithms is that we can construct the frequent sets of objects by adding objects to a set that is frequent until it is not frequent anymore. When objects are graphs, a graph is frequent if all its subgraphs are also frequent. In the general case this last proposition is not true but if we are in the context of disjoint node sets for subgraphs, this proposition is true. On Fig. 3, the graph c) has only one occurrence in the graph a). If we consider that subgraphs can have common nodes, three occurrences of graph b) can be found in graph a). In our case, nodes only participate in the representation of a single symbol. Hence, subgraphs must have distinct nodes. Then, only one occurrence of graph b) can be found graph a).

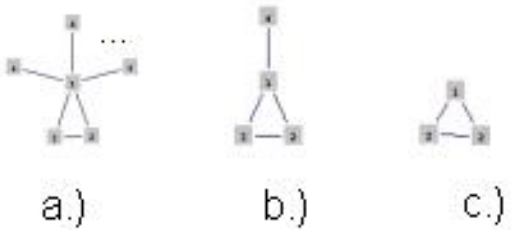


Fig. 3. Illustration for frequent subgraph search

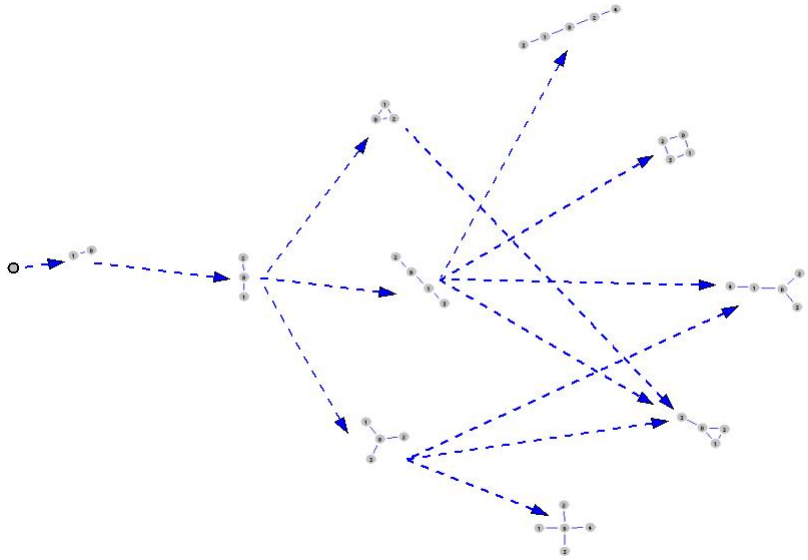


Fig. 4. Non-isomorphic graph network

In the algorithm used here, in order to reduce time complexity, we compute a network of non-isomorphic graphs off-line.

The network is used to guide the search for frequent subgraphs and to avoid isomorphism related computations (exponential in time) during this procedure. The network contains all graphs that have less than MAX edges. The graphs and their relations of inclusion are generated using the method presented in [15]. This method generates all non isomorphic subgraphs of a particular size. The complexity of this method is exponential.

Based on the relation of inclusion between these graphs the network is an acyclic oriented graph, whose nodes are all non-isomorphic graphs with less than MAX edges, where MAX is an input parameter. Fig. 4 presents how a search for frequent subgraphs is done. If at a certain stage a graph is not frequent, all of its descendants, with more edges, cannot be frequent. This network was computed with *MAX=9* in our application. Two reasons sustain this choice: the size of the network increases more than exponentially with the number of graph edges and the symbols are rarely expressed with graphs that have a bigger number of edges. The algorithm uses the information contained in the network of non-isomorphic graphs (the inclusion relations and automorphisms for each graph) to efficiently search for frequent subgraphs. Based on the non-isomorphic graph network, the search for frequent subgraphs is done in polynomial time.

2.1Algorithm

Network initialisation till level *MAX*

begin
Input An undirected labelled graph
Output A list of frequent subgraphs and for each one the apparition list
k:=1
while *k<=MAX*
for all graphs that can be frequent
let *G* be the current graph
using the apparition lists of his predecessors the apparition list of G is computed
if the apparition list contains more entries than a *threshold*
then graph *G* is considered frequent
if *G* is frequent

then update the list of predecessor setting the (inclusion in a frequent graph) flag on *true*
else update the successors of G setting the flag, for the possibility to be frequent, on *false*
for all frequent graphs from level *k-1*
update the list of apparitions taking into account the inclusion in other frequent graphs
update accordingly the frequent flag
k:=k+1
end while
end.

The threshold is computed using the following formula:

$$threshold = p * \min(\frac{e}{e'}, \frac{n}{n'}) \quad (1)$$

This formula represents an approximation of the maximum number of subgraphs that can be found in a graph. We consider that a subgraph is frequent if the number of occurrences is bigger than p% out of the maximum (possible) total number of subgraphs having e' edges and n' nodes. This algorithm can be applied to a graph or a set of graphs associated to a document or a collection of documents.

3 Rules and meta-rules

After some symbols were found using the above algorithm, relations between those symbols can be considered. The search for association rules between symbols is made using the "A priori" algorithm [7]. In the subsequent paragraphs the setting of this algorithm is presented. If we consider a set of symbols all having a common property, for example being on the same level in the inclusion tree (this tree models the inclusions between shapes), we may say this set of symbols participates in a transaction. All transactions are considered when relations between symbols are computed. An example for a set of transactions that describes how the objects are related can be:

$$T_1(o_1, o_2, o_3); T_2(o_1, o_2); T_3(o_2, o_3); T_4(o_1, o_2, o_4)$$

From this set of transactions one can extract a rule as the following "if the object *o1* participates in a transaction then the object *o2* will probably be there too".

The transactions can be defined using other criterions such as: a document represents a single transaction. The relations found have the meaning that if a set of symbols appears in a document then it is highly probable that the consequent set of symbols will appear as well.

In the single graph setting we can relate transactions to graph partitioning or subgraph clustering. However, in the present paper only transactions based on the inclusion relation are used.

Applying the A priori algorithm in this context (i.e. using the above described transactions) we find relations of the following type:

$$(o_{i1}, o_{i2}, \ldots, o_{in}) \Rightarrow (o_{j1}, o_{j2}, \ldots, o_{jm}) \quad (2)$$

Where

$$(o_{i1}, o_{i2}, \ldots, o_{in}) \cap (o_{j1}, o_{j2}, \ldots, o_{jm}) = \varnothing$$

If we consider a rule R obtained by the "A priori" algorithm, we can compute for each transaction whether R is confirmed or not. The confirmation is verified using the logical definition of the implication relation.

This computation has the following meaning: a rule is considered in its own right as a pattern and we consider that this particular rule appears in the transaction if it is confirmed in that transaction.

When in a given document we find a relation between some symbols then this fact implies the existence of a relation between some other symbols in the document.

Considering rules as patterns can be recursively applied in order to obtain meta-rules of type:

$$((o_{i1},...,ok_1) \Rightarrow (o_{i2},...,ok_2)) \Rightarrow ((o_{i3},...,ok_3) \Rightarrow (o_{i4},...,ok_4)) \quad (3)$$

or

$$(o_{i1},...,ok_1) \Rightarrow ((o_{i2},...,ok_2) \Rightarrow (o_{i3},...,ok_3))$$

or

$$((o_{i1},...,ok_1) \Rightarrow (o_{i2},...,ok_2)) \Rightarrow (o_{i3},...,ok_3)$$

The meta-rules found add knowledge to the associations and are not equivalent with simple rules. To support this assertion, we present an example where a meta-rule is not reducible to a simple rule (like Eq. 2.). The meta-rule $(o_1 \Rightarrow o_2) \Rightarrow (o_3 \Rightarrow o_4)$ is written in a disjunctive normal form as: $\bar{o}_1 o_2 + \bar{o}_3 + o_4$ but no simple rule such as $(o_1, o_2) \Rightarrow (o_3, o_4)$ or $o_1 \Rightarrow (o_2, o_3, o_4)$ written in a disjunctive normal form will contain a conjunction of a statement letter and a negation of other letter as it is the case for the meta-rule.

These types of meta-rules are more difficult to be expressed in informal language but are closer to the domain knowledge rules. One can describe a relation $R_1 \Rightarrow R_2$ between rules as follows: all transactions that contain a certain rule will probably contain the second rule as well.

4 Examples

4.1 Tutorial example

This section presents a didactic example of our approach applied on a synthetic document (Fig. 5.) containing architectural symbols. First, connected components, loops and neighbouring relations are extracted. After that, the neighbouring graph is built (Fig. 6(a)). Inclusion of shapes can be obtained from the graph [17]. Then, the corresponding inclusion tree is obtained (Fig. 6(b)). The threshold s is computed (s = 6) by applying equation (1) with p = 0.2. Then a subgraph is considered frequent if we can find 6 occurrences at least. The results of frequent subgraph search are shown on Fig. 7. In this search the inclusion relation is not considered as a neighbouring relation. Using the discovered symbols, transactions that contain these symbols can be obtained. Each transaction represents a leaf of the inclusion tree.

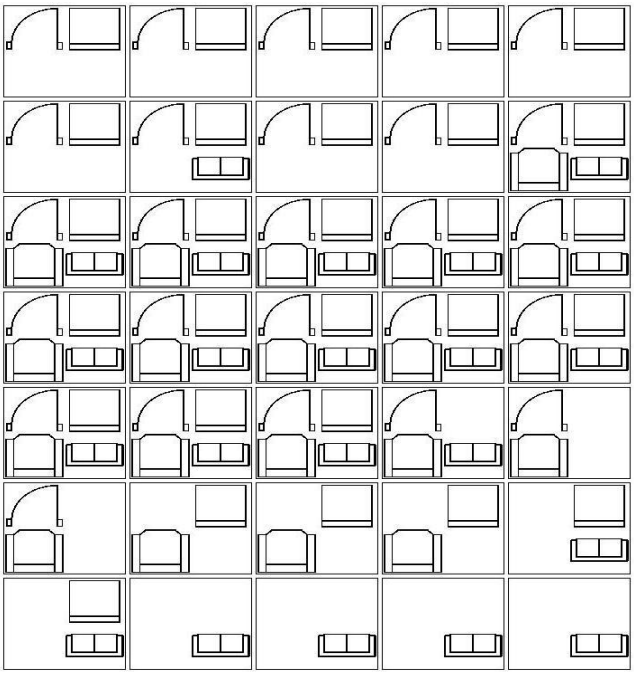


Fig. 5. A technical drawing

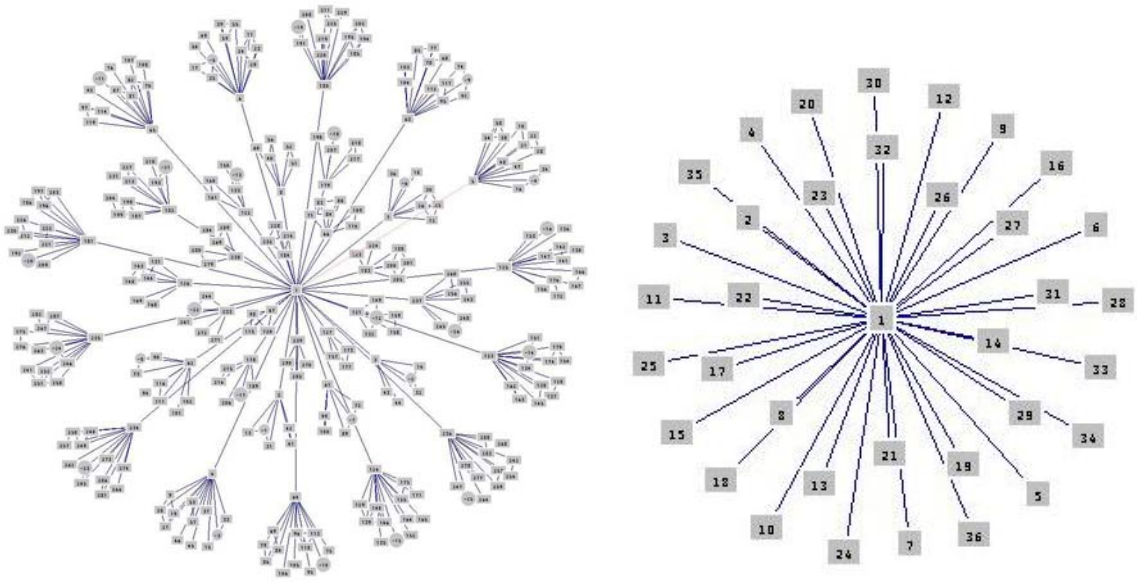


Fig. 6. Neighbourhood graph and inclusion tree

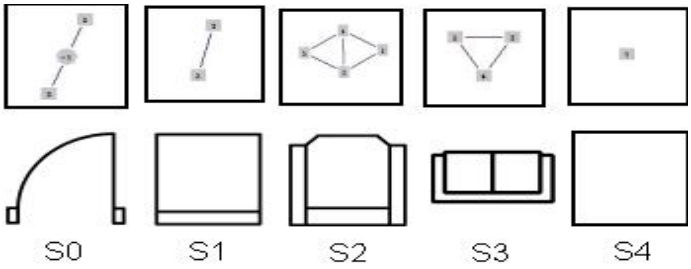


Fig. 7. Frequent subgraphs and corresponding symbols

The symbols are named S_0, S_1, S_2, and S_3. Considering the above assumptions the transactions are:

$T_1(S_0,S_1),T_2(S_0,S_1),T_3(S_0,S_1),T_4(S_0,S_1),T_5(S_0,S_1),T_6(S_0,S_1),$
$T_7(S_0,S_1,S_3),T_8(S_0,S_1),T_9(S_0,S_1),T_{10}(S_0,S_1,S_2,S_3),T_{11}(S_0,S_1,S_2,S_3),T_{12}(S_0,S_1,S_2,S_3),$
$T_{13}(S_0,S_1,S_2,S_3),T_{14}(S_0,S_1,S_2,S_3),T_{15}(S_0,S_1,S_2,S_3),T_{16}(S_0,S_1,S_2,S_3),T_{17}(S_0,S_1,S_2,S_3),$
$T_{18}(S_0,S_1,S_2,S_3),T_{19}(S_0,S_1,S_2,S_3),T_{20}(S_0,S_1,S_2,S_3),T_{21}(S_0,S_1,S_2,S_3),T_{22}(S_0,S_1,S_2,S_3),$
$T_{23}(S_0,S_1,S_2,S_3),T_{24}(S_0,S_2,S_3),T_{25}(S_0,S_2),T_{26}(S_0,S_2),T_{27}(S_1,S_2),T_{28}(S_1,S_2),T_{29}(S_1,S_2),$
$T_{30}(S_1,S_3),T_{31}(S_1,S_3),T_{32}(S_3),T_{33}(S_3),T_{34}(S_3),T_{35}(S_3).$

The support and the confidence are often used to qualify association rules. For a rule $a \Rightarrow b$, these are defined by:

$$Support=\frac{n_a}{n} \qquad Confidence=\frac{n_{ab}}{n_a}$$

where n is the number of transactions, n_a is the number of transactions which satisfy a and n_{ab} is the number of transaction which satisfy $a \wedge b$.

Based on these transactions the following rules and meta-rules were obtained:

R_1:$(S_0 \Rightarrow S_1)$ *support=0.74 confidence=0.88*

R_2:$(S_2 \Rightarrow S_0)$ *support=0.57 confidence=0.85*

R_3:$(S_3 \Rightarrow (S_2 \Rightarrow S_0))$ *support=0.62 confidence=1.0*

The rules were found considering a threshold of *0.8* for confidence and *0.5* for support in the "A priori" algorithm.

The meta-rule found using the above thresholds has a significance (in the context of these artificially created document image) equivalent with a logo in a real document image. When we find a certain logo we expect rules between symbols which are specific to that document.

4.2 Robustness

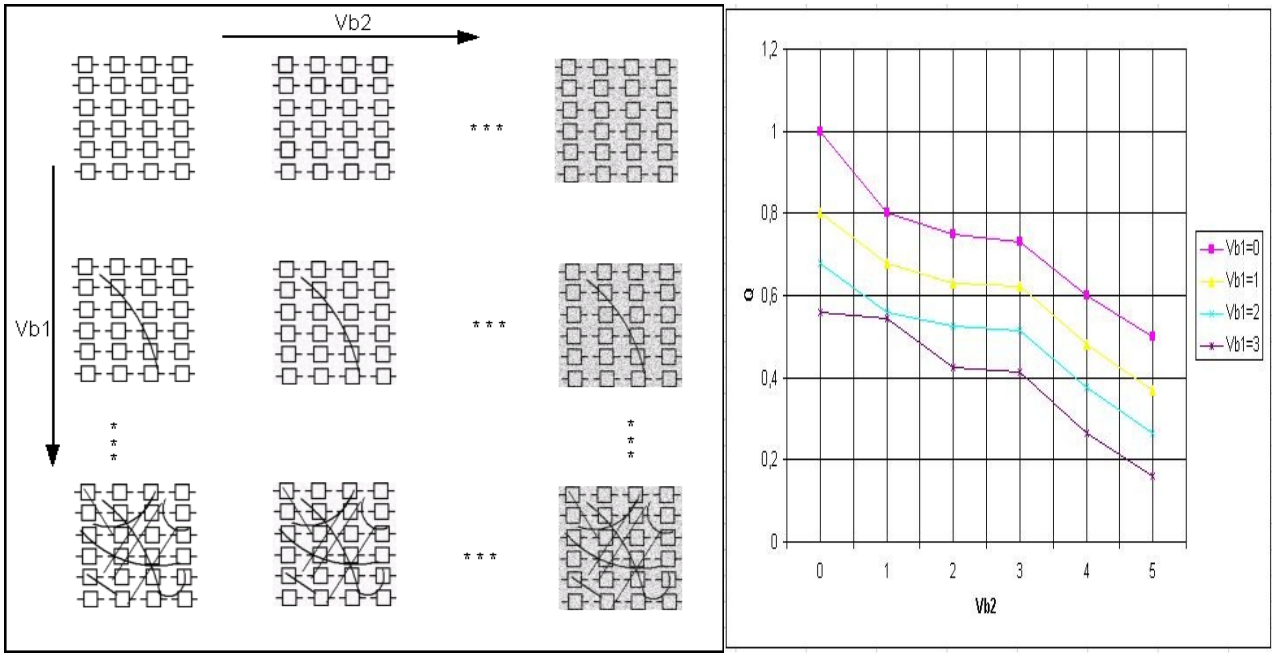
This section presents an experiment which aims at assessing the robustness of our approach. Fig. 8(a) represents several occurrences of the same symbol with different levels of noise. Two kinds of noise have been introduced :

– *Vb1* models the connectivity of several graphic information,

– *Vb2* is a gaussian noise on the grey level image.

The Vb1 noise highlights the capacity of the method to deal with connected and distorted symbols. Even when some symbols are unrecognisable the property of being frequent is kept.

Fig. 8(b) gives for each noise level of *Vb1*, the proportion of found symbols in relation to *Vb2*. Even if this proportion decreases with the noise, our objective is not to extract all symbols but rather to find redundancies that qualify the document. However, we can conclude that the thresholds have to be adapted to the noise on the document image.



(a)Different noise levels (b)Robustness evaluation

Fig. 8. Robustness to noise

5 Conclusions

The research undertaken represents a novel approach for finding symbols in line drawing documents as well as for discovering relations between automatically mined symbols. The approach uses data mining concepts for knowledge extraction. It aims at finding frequent symbols and relations. These frequent patterns are part of the document model and can be put in relation with the domain knowledge. The exposed method can be applied to other graph representations of a document. The only condition is that the document graph should contain symbols as disjoint graphs. In our future works, we will apply this approach to layout structures of textual document images to extract formatting rules. Some follow-up activities could be:

– post-processing of the neighbourhood graph in order to attenuate the noise influence;

– employment of error tolerant graph matching;

– utilization, at a semantic level, of more powerful indices for association rules;

– creation of a hierarchy of rules, probably a similar approach with Gras et al. [17].

References

[1] Yan, X., Han, J.: “Closegraph: mining closed frequent graph patterns”. In: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM Press (2003) 286–295

[2] Altamura, O., Esposito, F., Malerba, D.: “Transforming paper documents into xml format with Wisdom++”. *International Journal on Document Analysis and Recognition* 4 (2001) 2–17

[3] Messmer, B.: "*Efficient Graph Matching Algorithms for Preprocessed Model Graphs*". PhD thesis, University of Bern, CH, Institute of Applied Mathematics (1995)

[4] Berardi, M., Ceci, M., Malerba, D.: "Mining spatial association rules from document layout structures". In: *Proceedings of the Third International Workshop on Document Layout Interpretation and its Applications*. (2003)

[5] Cornuéjols, A., Mary, J., Sebag, M.: « Classification d'images à l'aide d'un codage par motifs fréquents ». In: *Actes de la Journée analyse de données, statistique et apprentissage pour la fouille d'image du Congrès RFIA*. (2004) 11–16

[6] Ordonez, C., Omiecinski, E.: "Discovering association rules based on image content". In: *Proceeding of the IEEE Advances in Digital Libraries Conference*. (1999)

[7] Agrawal, R., Srikant, R.: "Fast algorithms for mining association rules". In Bocca, J.B., Jarke, M., Zaniolo, C., eds.: *Proc. 20th Int. Conf. Very Large Data Bases*, VLDB, Morgan Kaufmann (1994) 487–499

[8] Washio, T., Motoda, H.: "State of the art of graph-based data mining". *SIGKDD Explor. Newsl*. 5 (2003) 59–68

[9] Kuramochi, M., Karypis, G.: "Frequent subgraph discovery". In: *Proceedings of the International Conference on Data Mining*. (2001)

[10] Inokuchi, A., Washio, T., Motoda, H.: "An apriori-based algorithm for mining frequent substructures from graph data". In: *Proceedings of the Conference on Principle and Practice of Knowledge Discovery in Databases*. (2000)

[11] Gupta, A., Jain, R.: Visual information retrieval. *Comm. Assoc. Comp. Mach.*, 40 (May 1997) 70-79

[12] Barbar D., Domeniconi C., Kang N., Classifying Documents Without Labels, In : *Proceedings of the Fourth SIAM International Conference on Data Mining*, Lake Buena Vista, Florida, USA, April 22-24,2004

[13] Khotanzad, A. and Hong, Y.H. Invariant Image Recognition by Zernike Moments. IEEE Trans. on PAMI, 12 (5). 289-497, 1990

[14] Milligan, G. W., Cooper, M.C.: An Examination of Procedures for Determining the Number of Clusters in a Data Set. Psychometrika, 58(2),(1985)159-179.

[15] Skvoretz J., An algorithm to generate connected graphs, In: *Current research in social psychology*, Vol. 1, No. 5, 1996

[16] Pavlidis, T., Algorithms or Graphics and Image Processing, Computer Science Press, 1982.

[17] Gras, R., Kuntz, P., Briand, H.: « Hiérarchie orientée de règles généralisées en analyse implicative ». In: *Actes des journées francophones d'extraction et de gestion des connaissances*. (2003)