Using Bags of Symbols for Automatic Indexing of Graphical Document Image Databases

Eugen Barbu, Pierre Héroux, Sébastien Adam, Eric Trupin

Using Bags of Symbols for Automatic Indexing of Graphical Document Image Databases

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

LITIS, Université de Rouen,
F-76800 Saint-Etienne du Rouvray, France
Eugen.Barbu@univ-rouen.fr

Abstract. A database is only usefull if it is associated a set of proce- dures allowing to retrieve relevant elements for the users' needs. A lot of IR techniques have been developed for automatic indexing and retrieval in document databases. Most of these use indexes depending on the tex- tual content of documents, and very few are able to handle graphical or image content without human annotation.

This paper describes an approach similar to the bag of words technique for automatic indexing of graphical document image databases and diferent ways to consequently query these databases. In an unsupervised manner, this approach proposes a set of automatically discovered sym- bols that can be combined with logical operators to build queries.

1 Introduction

A document image analysis (DIA) system transforms a document image into a description of the set of objects that constitutes the information on the document in a way that can be processed and interpreted by a computer [1]. Documents can be classified in mostly graphical or mostly textual documents [2]. The mostly textual documents also known as structured documents respect a certain layout and powerful relations exist between components. Examples of such documents are technical papers, simple text, newspapers, program, listing, forms. . . Mostly graphical documents do not have strong layout restrictions but usually relations exist between diferent document parts. Examples of this type of documents are maps, electronic schemas, architectural plans. . .

For both categories of documents, graph based representations can be used to describe the image content (e.g. region adjacency graph [3] for graphical and Voronoi-based neighbourhood graph [4] for textual document images).

This paper presents an approach similar with the "bag of words" method from Information Retrieval (IR) field applied to graphical document images. A ocument representation is built based on a bag of symbols found automatically using graph mining [5] techniques. In other words, we consider as "symbols", the frequent subgraphs of a graph-based document representation and we investigate if the description of a document as a bag of "symbols" can be profitably used in an indexing and retrieval task.

The approach has the ability to process document images without knowledge of models for document content. Frequent items are used in clustering of textual documents [6], or in describing XML documents [7], but we do not know any similar approach in the DIA field.

In the area of research for document image indexing, approaches based on partial document interpretation exist [8]. The images are automatically indexed using textual and graphical cues. The textual cues are obtained from the results proposed by an OCR system. The graphical indices are obtained by user anno- tation, or by an automatic procedure. In [9], Lorenz and Monagan present an automatic procedure. Junctions of adjacent lines, parallel lines, collinear lines and closed polygons are used as image features for indexing. Then, a weighting schema is used to reflect the descriptive power of a feature. In our paper, we also use term weighting but on a representation from a higher semantic level than the simple features used in [9].

The outline of this paper is as follows. Section 2 presents the graph represen- tation used and shows how we create this representation from a document image. Section 3 presents the graph-mining method used. In Sect. 4, we describe how we search documents based on dissimilarities between bags of objects. Section 5 shows experimental results. We conclude the paper and outline perspectives in Sect. 6.

2 Graph Representation

Eight levels of representation for document images are proposed in [10]. These levels are ordered according to their aggregation relations. Data array, primitive, lexical, primitive region, functional region, page, document, and corpus level are the representation levels proposed.

Without loosing generality, in the following paragraphs we focus on a graph- based representation build from the primitive level. The primitive level contains objects such as connected components (set of adjacent pixels with the same color) and relations between them. From a binary (black and white) document image we extract connected components. The connected components are represented by graph nodes. On each connected component we extract features. In the cur- rent implementation, the extracted characteristics are rotation and translation invariant features based on Zernike moments [11]. These invariants represent the magnitudes of a set of orthogonal complex moments of a normalized image.

Let I be an image and $C(I)$ the connected components from I, if $c \in C(I)$, c is described as $c = (id, P)$, where id is a unique identifier and P the set of pixels the component contains. Based on this set P , we can compute the center for the connected component bounding box and we can also associate a feature vector to it. Based on that, $c = (id, x, y, v)$, $v \in R^n$. Subsequently, using a clustering procedure on the feature vectors, we can label the connected component and reach the description $C = (id, x, y, l)$ where l is a nominal label. The graph $G(I)$ representing the image is $G = G(V(I), E(I))$. Vertices $V(I)$ correspond to connected components and are labelled with component labels. An edge between vertex u and vertex w exists if and only if $(u.x - w.x)^2 + (u.y - w.y)^2 < t$, where t is a threshold that depends on the global characteristics of image I (size, number of connected components,. . .).

The following paragraph presents the clustering procedure used to associate a label to each connected component.

Clustering methods can by categorized into partitional and hierarchical tech- niques. Partitional methods can deal with large sets of objects ("small" in this context means less than 300) but needs the expected number of clusters in input. Hierarchical methods can overcome the problem of number of clusters by using a stopping criterion [12] but are not applicable on large sets due to their time and memory consumption.

In our case the number of connected components that are to be labelled can be larger than the limit of applicability for hierarchical clustering methods. On the other hand, we cannot use a partitional method because we do not know the expected number of clusters. Based on the hypothesis that a "small" sample can be informative for the geometry of data, we obtain in a first step an estimation for the number of clusters in data. This estimation is obtained using an ascendant clustering algorithm with a stopping criterion. The number of clusters found in the sample is used as input for a partitional clustering algorithm applied on all data.

We tested this "number of cluster estimation" approach using a hierarchical ascendant clustering algorithm [13] that employs Euclidean distance to com- pute the dissimilarity matrix, complete-linkage to compute between-clusters dis- tances, and Calinsky-Harabasz index [12] as a stopping criterion. The datasets (T_1, T_2, T_3) (see Table 1) are synthetically generated and contain well separated (not necessary convex) clusters.

Table 1. Data sets description

T	$/T/$	number of clusters
T_1	24830	5
T_2	32882	15
T_3	37346	24

Considering S the sample extracted at random from a test set, in Table 2, we present predicted cluster numbers obtained for diferent sample sizes. After repeating the sampling procedure several times, we obtain a set of estimations for the number of clusters. We can see that by using a majority voting decision rule we can find the good number of clusters in most of the cases and even when the sample size is very small (50 or 100) compared to the data set size.

We used our sampling approach combined with the k-medoids clustering algo- rithm [14] on the connected components data set from images in our corpus (see Sect. 5). The k-medoids clustering algorithm is a more robust version of the well known k-means algorithm. The images from our corpus contain 6730 connected components. The proposed number of clusters using ten samples of size 600 is [16,14,17,16,16,19,7,17,15,16] and by considering the majority voting, we use 16 clusters as input to the partitional clustering algorithm.

Table 2. Proposed cluster numbers

|S|

50
100
300
500
600
700

T_1
7,
9,
7,
5,
5,
7,

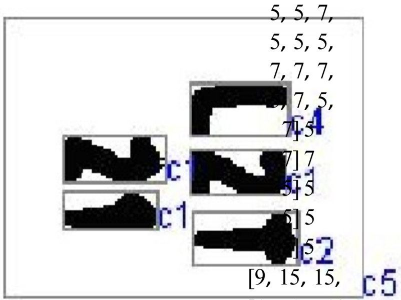
[6, 8,
[5, 7,
[7, 5,
[8, 7,
[5, 5,
[5, 5,

6, 5, 6,

7, 5, 5,
8, 7, 5,
5, 5, 5,
5, 5, 7,
5, 7, 5,

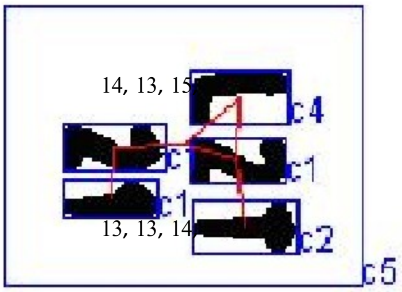
6, 6, 5,

7, 5, 5,
5, 5, 7,
5, 5, 5,
7, 7, 7,
5, 7, 5,
7] 5
7] 7
5] 5
5] 5
[illegible]] 5

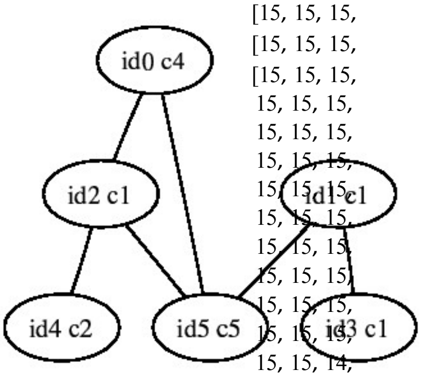


T_2

[9, 15, 15,
[15, 15, 13,
[15, 15, 15,
[15, 15, 15,
[15, 15, 15,
[15, 15, 15,



14, 13, 15
13, 13, 14



15, 15, 15,
15, 15, 15,
15, 15, 15,
15, 15, 15,
15, 15, 15,
15, 15, 15,
15, 15, 15,
15, 15, 15,
15, 15, 15,
15, 15, 14,

15] 15

15] 15
14] 15
15] 15
15] 15
15] 15

T_3

[11, 7, 9,
[6, 14, 23,
[22, 24, 23,
[21, 25, 25,
[20, 25, 21,
[23, 20, 21,

18, 7, 7,

21, 7, 17,
19, 23, 24,

6, 4, 14,

8] 7

24, 22, 24,
24, 19, 23,
20, 25, 24,
23, 16, 12,
21, 21, 24,
23, 24, 24,
24, 25, 24,
24, 21, 25,
11] 23
24] 24
24] 24
22] 24
24] 24

After labelling the connected components (nodes in the graph), we now de- scribe the way these nodes are linked. The edges can be labelled or not (if unla- beled, the significance is Boolean: we have or don't have a relation between two

(a) Initial image (b) Connected components labelling

(c) Graph construction (d) Graph transaction

Fig. 1. An image and its associated graph transaction

connected components) and there can be relations of spatial proximity, based on "forces" [15], orientation or another criterion. In our current implementation the distance between centers of connected components is used (see Fig. 1). If the distance between two connected component centers is smaller than a threshold, then an edge will link the two components (nodes).

3 Graph Mining

"The main objective of graph mining is to provide new principles and efcient algorithms to mine topological substructures embedded in graph data" [5].

Mining frequent patterns in a set of transaction graphs is the problem of finding in this set of graphs those subgraphs that occur more times in the trans- actions than a threshold (minimum support). Because the number of patterns can be exponential, the complexity of this problem can also be exponential. An approach to solve this problem is to start with finding all frequent patterns with one element. Then, these patterns are the only candidates among which we search for frequent pattens with two elements, etc. in a level-by-level setting. In order to reduce the complexity, diferent constraints are used: the minimum support, the subgraphs are connected, and do not overlap.

The first systems emerged from this field are SUBDUE and GBI [5]. These approaches use greedy techniques and hence can overlook some patterns. The SUBDUE system searches for subgraphs in a single graph using a minimum de- scription length-based criterion. Complete search for frequent subgraphs is made in an ILP framework by WARMR [5]. An important concept is that of maximal subgraph. A graph is said to be maximal if it does not have a frequent super- graph [16]. The graph-mining systems were applied to scene analysis, chemical components databases and workflows. A system that is used to find frequent pat- terns in graphs is FSG (Frequent Subgraph Discovery) that "finds patterns cor- responding to connected undirected subgraphs in an undirected graph database" [17].

In our document image analysis context we are interested in finding maximal frequent subgraphs because we want to find symbols but to ignore their parts.

The input for the FSG program is a list of graphs. Each graph represents a transaction. FSG is efective in finding all frequently occurring subgraphs in datasets containing over 200,000 graph transactions [17]. We present sub- sequently how we construct the transaction list starting from a set of document images. Using the procedure presented in Sect. 2, we create for each document an undirected labelled graph.

Every connected component of this graph represents a transaction. We can further simplify the graphs by removing vertices that cannot be frequent and their adjacent edges. Using FSG we extract the frequent subgraphs and we con- struct a bag of graphs occurring in each document. In the following paragraphs, we consider that the frequency condition is sufcient for a group of connected components to form a symbol and we will conventionally make an equivalence between the frequent subgraphs found and symbols. As we can see in the example (Fig. 2), the proposed symbols are far from being perfect due to the image noise, connected components clustering procedure imperfections. . . however we can notice the correlation between this artificial symbol and the domain symbols.

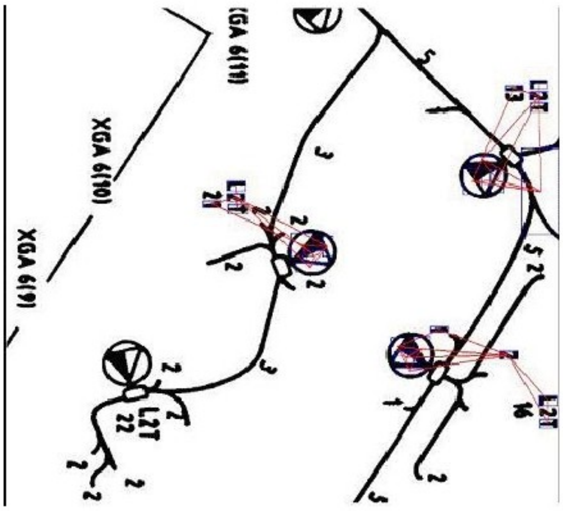


Fig. 2. Occurences of a frequent subgraph in an image

In conclusion, the subgraphs proposed as frequent are used to model a docu- ment as a bag of symbols. Because some documents may not contain any sym- bols, the document representation is based on two vectors containing connected components labels, and symbols labels.

$$A : (c_1, c_2, . . . , c_n) , (s_1, s_2, . . . , s_m)$$

where c_i is the number of connected components labelled as i and s_j is the number of occurences of symbol j in document A.

4 Dissimilarity Between Document Descriptions

In this paragraph, we present the measure employed to qualify the dissimilarity between the descriptions of two document images.

A collection of documents is represented by a symbol-by-document matrix A, where each entry represents the occurrences of a symbol in a document image, $A = (a_{ik})$, where a_{ik} is the weight of symbol i in document k. Let f_{ik} be the frequency of symbol i in document k, N the number of documents in the collection, and n_i the total number of times symbol i occurs in the whole collection. In this setting, according to [18], one of the most efective weighting scheme is entropy-weighting. The weight for symbol i in document k is given by:

$$a_{ik} = \log (1 + f_{ik}). \left(1 + \frac{1}{\log N} \sum_{j=1}^{n} \frac{f_{ij}}{n_i} \log \frac{f_{ij}}{n_i}\right)$$

Now, considering two documents A and B with the associated weights $A = (a_1, a_2, \ldots, a_t)$, $B = (b_1, b_2, \ldots, b_t)$ where t is the total number of symbols, then

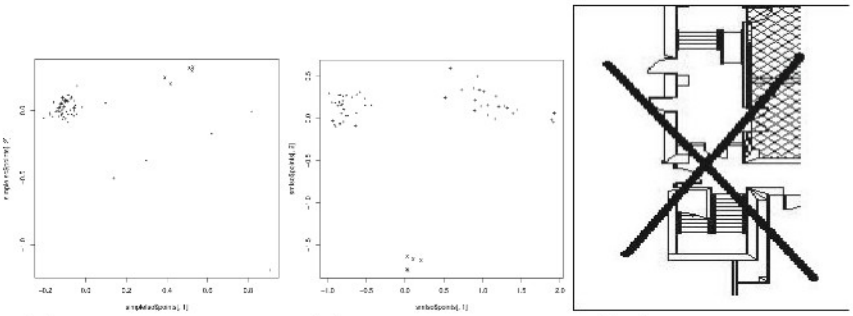
$$d(A, B) = 1 - \frac{\sum_{i=1}^{t} a_i.b_i}{\sqrt{\sum_{i=1}^{t} a_i^2 . \sum_{i=1}^{t} b_i^2}}$$

represents a dissimilarity measure based on the cosine correlation.

5 Experiments

The corpus used for evaluation contains 60 images from 3 categories: electronic (25 images) and architectural schemas (5 images) and engineering maps (30 images) (see Fig. 3). In order to present a corpus summary we employed a mul- tidimensional scaling algorithm to represent in a two dimensional plot the dissim- ilarities between documents (see Fig. 4). Each document image is described with one of the following types of features: simple density and surface based charac- teristics (a vector with 30 components) or the connected components and symbol lists described above. In Fig. 4(a) we present the dissimilarities between images represented using simple features. In Fig. 4(b) are plotted the dissimilarities be- tween the document images computed using the cosine correlation presented in Sect. 4. The engineering maps are plotted using '*' symbols, electronic schemas with '+' , and the architectural schemas with 'x'.

We further test the two representations in a classification context. Using a 10 fold stratified cross validation procedure and John C. Platt's sequential minimal optimisation algorithm for training a support vector classifier [19], we obtained the following results given in Tab. 3



(a) Engeneering maps (b) Electronic schema (c) Architectural plan

Fig. 3. Corpus images

Table 3. Classification results using the simple characteristics

Features	number of correctly classification classified instances	rates
only simple features		55
		91.67
% only bag of symbol representation		57
		95 %
simple features and bag of symbol %		58 96.67

We can see in Fig. 4 and the classification results (3) that the bag of symbols representation allows a better separation beetween image classes. This fact has an important influence on the quality of the query results.

A query can be an image, a list of symbols and connected components, or only one of the later lists.

query: $(c_1, c_2, \ldots, c_n)$, $(s_1, s_2, \ldots,$

$s_m)$

query: $(s_1, s_2, \ldots, s_m)$

query: $(c_1, c_2, \ldots, c_n)$

After using the graph mining algorithm on the presented corpus we obtain 52 frequent subgraphs. This subgraphs are the symbols that will be used in queries, and are numbered from 1 to 52. The description of the first 4 documents (in terms of what symbols and what are their corresponding frequencies) is subsequently presented :

$(s_1, 1)(s_2, 2)(s_3, 3)(s_4, 1)(s_5, 4)(s_6, 3)(s_7, 2)(s_8, 2)(s_9, 2)(s_{13}, 1)(s_{14}, 1)(s_{16}, 3)$
$d_1 : (s_{17}, 2)(s_{18}, 1)(s_{19}, 4)(s_{20}, 6)(s_{21}, 1)(s_{22}, 4)(s_{23}, 2)(s_{24}, 4)(s_{25}, 2)(s_{26}, 2)$
$(s_{35}, 1)(s_{36}, 1)(s_{37}, 1)(s_{41}, 1)(s_{45}, 1)(s_{46}, 1)(s_{49}, 1)(s_{51}, 1)(s_{52}, 1)$

$d_2 : (s_1, 1)(s_2, 3)(s_3s3),(s_4,s2)(,s1),(2)(s,_61)1)s(s_7,,1)((ss_{16},,1)\ (s_{19}, 1)(s_{20}, 3)$

$(s_{22}, 1)$

, ^{5} , 1) ^{48} 2)
$(s_{23}, 2)(s_{25}, 2)(_{39}\ 1)(_{42}$ $s_{43}\ (_{47}$

$d_3 : (s_1, 1)(s_2, 1)(s_3s4),(s_4,s1)(,s5),(1)$

$(s,_81)1)s(s_{11}1)1)s(s_{12}1)1)s(s_{13}1)1)s(s_{16}1)1)(s_{19}, 2)$

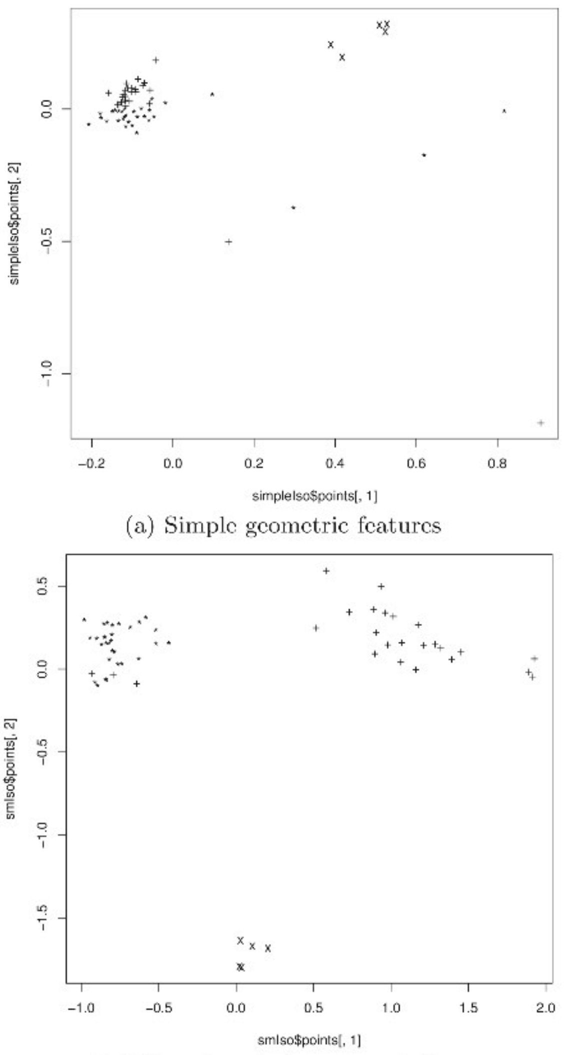
, ^{7}
,
,
,
,
,
$(s_{20}, 4)(s_{21}, 1)(_{22}\ 3)(_{25}$ $s_{35}\ (_{39}, (_{47}, (_{48}, (_{52},$

$(s_1, 4)(s_2, 4)(s_3, 3)(s_4, 2)(s_5, 2)(s_6, 3)(s_7, 1)(s_8, 2)(s_9, 2)(s_{11}, 3)(s_{12}, 1)(s_{16}, 1)$

$d_4 : (s_{18},, 1)(s_{19},, 4)(s_{20},, 4)(s_{21},, 2)(s_{22},, 1)(s_{23},, 3)(s_{24},, 4)(s_{25},, 2)(s_{26},, 2)$

$(s_{36},, 1)$ $(s_{37}\ 1)(s_{39}\ 2)(s_{40}\ 2)(s_{41}\ 1)(s_{42}\ 1)(s_{44}\ 1)(s_{46}\ 1)(s_{47}\ 2)(s_{48}\ 2)(s_{49}\ 1)$
$(s_{51}, 2)(s_{52}, 2)$

In order to extract the formal description of a given query image we label the connected components of the query image, construct the graph, and em- ploy graph matching to detect which symbols occur in the query image. At the end of this process the query image is described by the two lists of connected components and symbols.



(a) Simple geometric features

(b) Bag of symbols representation

Fig. 4. Document representations presented in a two dimensional space with respect to their reciprocal dissimilarities

In order to evaluate experimental results we used precision and recall mea- sures. If A is the set of relevant images for a given query, and B is the set of retrieved images then :

$$\text{precision} = \frac{|A \cap B|}{|B|} \quad \text{recall} = \frac{|A \cap B|}{|A|}$$

As shown on Fig. 3, the corpus contains images that are scanned and contain real and artificial noise.

Table 4. Examples of queries and results

query	answer to query

(s_1, 4)	d_{37} dissimilarity=0.6782
	d_{15} dissimilarity=0.7843 d_4
	dissimilarity=0.8070 d_{13}
	dissimilarity=0.8450 d_{27}
	dissimilarity=0.8452

(s_1, 4)(s_2, 4)(s_3, 3)(s_4, 2)	d_2 dissimilarity=0.4233
	d_7 dissimilarity=0.4722 d_{22}
	dissimilarity=0.4864 d_{25}
	dissimilarity=0.5046 d_{14}
	dissimilarity=0.5054

(s_1, 1)(s_2, 3)(s_3, 3)(s_4, 2)(s_5, 2) d_2 : (s_6, 1)(s_7, 1)(s_{16}, 2)(s_{19}, 1)(s_{20}, 3)1) d_{22} (s_{23}, 2)(s_{25}, 2)(s_{39}, 1)(s_{42},	d_2 dissimilarity=0.0065 d_{25} dissimilarity=0.1949 dissimilarity=0.2136 (s_{22}, 1)
(s_{43}, 1)(s_{47}, 1)(s_{48}, 1)	d_{26} dissimilarity=0.2241 d_{21} dissimilarity=0.2362

Table 5. Queries recall and precision

	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9	Q_{10}
recall	0.75	0.5	0.48	0.55	0.56	0.76	0.6	0.4	0.32	0.16
precision	0.6	0.31	0.8	0.73	0.87	0.95	0.88	0.5	0.42	0.4

Table 4 gives 5 most relevant documents relative to the query.

Table 5 gives the recall and precision for 10 diferent queries. Queries Q1-4 represents symbol queries, i.e. as input is a list of symbols. The other queries are document images.

6 Conclusion

The research undertaken represents a novel approach for indexing document images. Our approach uses data mining techniques for knowledge extraction. It aims at finding image parts that occur frequently in a given corpus. These frequent patterns are part of the document model and can be put in relation with the domain knowledge.

Using the proposed method we reduce in an unsupervised manner the semantic gap between a user representation for a document image and the indexation system representation.

The exposed method can be applied to other graph representations of a doc- ument. In the near future, we will apply this approach to layout structures of textual document images.

Another follow up activity is to quantify the way noise afects the connected components labelling, and the manner in which an incorrect number of clusters can afect the graph mining procedure. Based on this error propagation study we can further improve our method. Other possible improvements can be obtained if we would use a graph-based technique that can deal with error tolerant graph matching.

References

1. Antonacopoulos, A.: Introduction to Document Image Analysis. (1996)
2. Nagy, G.: Twenty years of document analysis in pami. IEEE Trans. on Pattern Analysis and Machine Intelligence 22(1) (2000) 38-62
3. Pavlidis, T.: Algorithms for Graphics and Image Processing. Computer Science Press (1982)
4. Bagdanov, A.D., Worring, M.: Fine-grained document genre classification using first order random graphs. In: Proc. of the sixth International Conference on Document Analysis and Recognition. (2001) 79-83
5. Washio, T., Motoda, H.: State of the art of graph-based data mining. SIGKDD Explor. Newsletter 5(1) (2003) 59-68
6. Fung, B.C.M., Wang, K., Ester, M.: Hierarchical document clustering using frequent items. In: Proc. of the SIAM Conference on Data Maining. (2003)
7. Termier, A., Rousset, M., Sebag, M.: Mining xml data with frequent trees. In: Proc. of DBFusion Workshop. (2002) 87-96
8. Doermann, D.: The indexing and retrieval of document images : A survey. Technical report, LAMP (1998)
9. Lorenz, O., Monagan, G.: Automatic indexing for storage and retrieval of line drawings. In SPIE, ed.: Storage and Retrieval for Image and Video Databases (SPIE). Volume 2420. (1995) 216-227
10. Blostein, D., Zanibbi, R., Nagy, G., Harrap, R.: Document representations. In: Proc. of the IAPR Workshop on Graphic Recognition. (2003)
11. Khotazad, A., Hong, Y.H.: Invariant image recognition by zernike moments. IEEE Trans. on Pattern Recogntion and Machine Analysis 12(5) (1990)
12. 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
13. Gordon, A.D.: Classification. 2nd edition edn. Chapman & Hall (1999)
14. Kaufmann, L., Rousseeuw, P.J.: Clustering by means of medoids. In Dodge, Y., ed.: Statistical Data Analysis based on the $L1$ Norm and Related Methods, Elsevier Science (1987) 405-416
15. Tabbone, S., Wendling, L., Tombre, K.: Matching of graphical symbols in line-drawing images using angular signature information. International Journal on Document Analysis and Recognition 6(2) (2003) 115-125
16. Yan, X., Han, J.: Closegraph: mining closed frequent graph patterns. In Press, A., ed.: Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2003) 286-295
17. Kuramochi, M., Karypis, G.: An efcient algorithm for discovering frequent subgraphs. IEEE Transactions on Knowledge Data Engeneering 16(9) (2004) 1038-1051
18. Dumais, S.T.: Improving the retrieval information from external ressources, behaviour research methods. Instrument and Computers 23(2) (1991) 229-236
19. Platt, J.: Fast training of support vector machines using sequential minimal optimization. In Schölkopf, B., Burges, C., Smola, A., eds.: Advances in Kernel Methods - Support Vector Learning, MIT Press (1998)