

Retrieval by Layout Similarity of Documents Represented with MXY Trees

Francesca Cesarini, Simone Marinai, and Giovanni Soda

Dipartimento di Sistemi e Informatica – Università di Firenze
Via S.Marta, 3 – 50139 Firenze – Italy
{cesarini,simone,giovanni}@dsi.unifi.it

Abstract. Document image retrieval can be carried out either processing the converted text (obtained with OCR) or by measuring the layout similarity of images. We describe a system for document image retrieval based on layout similarity. The layout is described by means of a tree-based representation: the Modified X-Y tree. Each page in the database is represented by a feature vector containing both global features of the page and a vectorial representation of its layout that is derived from the corresponding MXY tree. Occurrences of tree patterns are handled similarly to index terms in Information Retrieval in order to compute the similarity. When retrieving relevant documents, the images in the collection are sorted on the basis of a measure that is the combination of two values describing the similarity of global features and of the occurrences of tree patterns. The system is applied to the retrieval of documents belonging to digital libraries. Tests of the system are made on a data-set of more than 600 pages belonging to a journal of the 19th Century, and to a collection of monographs printed in the same Century and containing more than 600 pages.

1 Introduction

The document image database described in this paper is used to store books and journals in digital libraries. In the last few years the largest public libraries in the world have digitized and stored in electronic format collections of books and journals. These documents can be browsed and retrieved on the basis of meta-data describing their content. Most of this information (e.g. title, authors, publishing date) is manually selected from catalog cards. Sometimes the pages are processed with OCRs in order to allow the user to retrieve documents on the basis of their textual content. In this case the retrieval by (imprecise) text content is possible with techniques derived from Information Retrieval (IR). Few systems allow the user to retrieve pages by layout similarity, with approaches somehow similar to those applied in Content Based Image Retrieval (CBIR). Document image retrieval is a research field that lies at the borderline between classic IR [1] and CBIR [2]. Two recent papers [3,4] investigated past research and future trends in document image retrieval. Most work in document image retrieval has been based on the processing of converted text with IR-based techniques [3]. Fewer methods approached the retrieval by layout similarity, and

related approaches have been considered for document page classification. The retrieval by layout similarity is useful for locating some meaningful pages in unlabeled collections of scanned pages. For instance, if a query page corresponds to the beginning of a chapter, then the retrieved pages are likely to provide information on the chapters in a book or collection of books. Illustrations in a book can be retrieved by using a page with an image as a query page.

A general framework for document image retrieval has been proposed in [5]. The system allows users to retrieve documents on the basis of both global features of the page and features based on blocks extracted by layout analysis packages. Global features include texture orientation, gray level difference histogram, and color features. The block-based features use a weighted area overlap measure between segmented regions. More recently, the combination of global (page-level) and local features has been furtherly investigated for computing visual similarity between document images for page classification [6]. In the latter approach a fixed-size feature vector is obtained by extracting some specific features in the regions defined by a grid overlapped to the page. Similarly, the method discussed in [7] takes into account a fixed grid partitioning of the page and uses features computed from the textual zones. A grid-based approach to construct a feature vector computed from the density of connected components was considered also in [8]. In order to overcome some problems related to the choice of an optimal grid size, a page classification method based on an MXY tree decomposition of the page has been recently proposed in [9]. This method relies on a MXY tree [10] built from a segmented image where the blocks are obtained with a commercial OCR. The MXY tree is afterwards encoded into a fixed size vector that is used as input to an MLP-based classifier.

In the document image retrieval system described in this paper we use an MXY tree based document representation with an approach tightly related with the page classification proposed in [9]. The pages are first segmented with a layout analysis tool provided with an OCR package¹. Blocks extracted by this tool are afterwards arranged in an MXY tree describing the page layout. A DBMS is used in order to store relevant information of digitized books and also for maintaining the MXY tree of the page. At the end of a session of database population, appropriate feature vectors describing both the global features of the page and the MXY tree structure are stored in the database. During retrieval, a query by example approach is considered. To this purpose the user first selects one sample page by browsing the collection; afterwards a comparison of the query feature vector with vectors in the database is performed with an appropriate similarity measure, and retrieved documents are shown to the user.

The paper is organized as follows: in the next section we describe the method for document image retrieval that has been implemented in the system described in Section 3. Experimental results on a first data set containing more than 1200 pages are reported in Section 4, while concluding remarks are in Section 5.

¹ The OCR used is FineReader Engine 4.0.

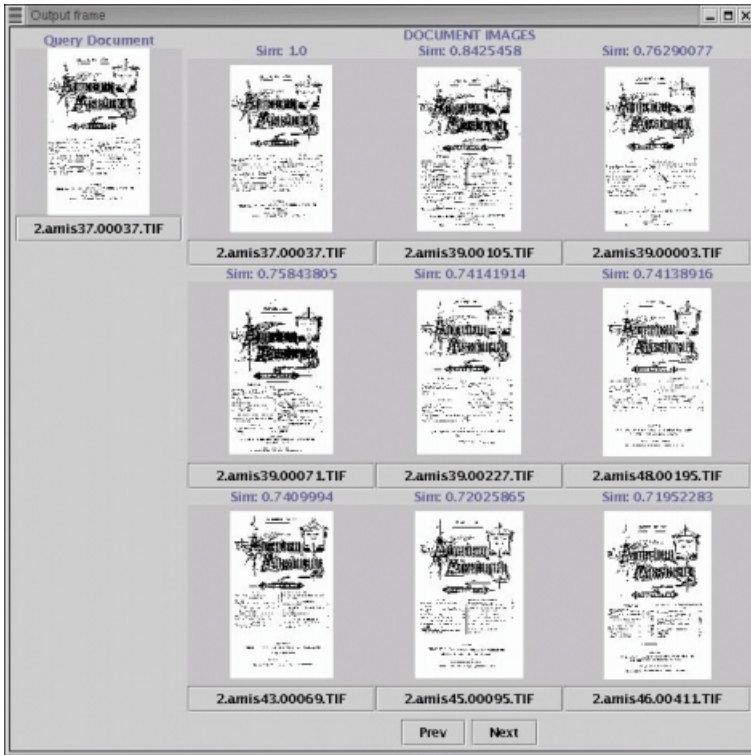


Fig. 1. User interface showing the result of the retrieval based on a query page shown on the left. The nine pages reported on the right are those with an higher similarity measure (reported over each image).

2 Document Image Retrieval

Document images are retrieved in our system by using a “query by example” approach: given a sample page the system computes the layout similarity of the page with all the pages in the database. Pages in the database are afterwards ranked on the basis of this similarity and shown to the user as system’s answer (Fig. 1). The page similarity is computed considering the distance between feature vectors describing the layout. Each feature vector contains two main groups of features. The first group contains global features describing the position and size of the printed part of the page with respect to the other pages belonging to the same book. The second group of features describes the layout of the page and is obtained from its MXY tree [10]. The page similarity is computed with an appropriate combination of two measures that operate independently for each group of features. In this section we describe the two parts of the feature vector and the similarity measure that we introduced in order to deal with this representation.

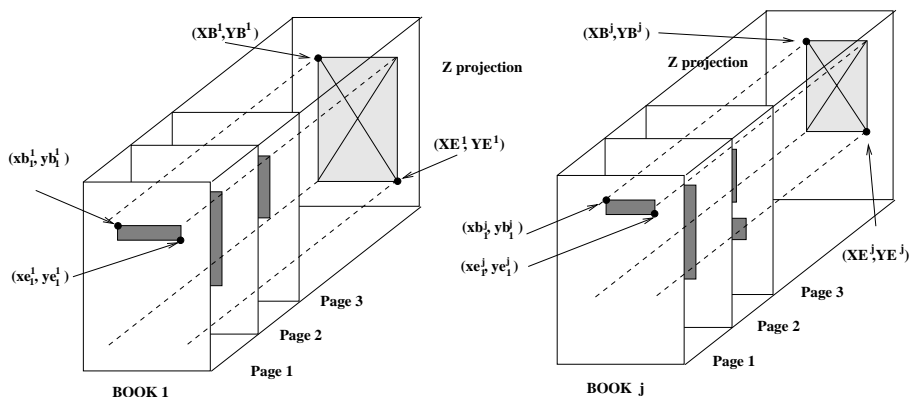


Fig. 2. Computation of the “book bounding box” by taking into account a Z -projection.

2.1 Global Features

The pages inserted in the database are first processed with layout analysis tools in order to extract homogeneous blocks. The layout is afterwards described by means of an MXY tree. This description has been demonstrated to be adequate for the classification of journal pages, where page layouts are quite complex and MXY trees are usually composed by several nodes [9]. When considering digitized books, there are some pages whose layout is made by a unique block. Typical examples are *regular pages*, corresponding to pages containing continuous text (the narrative part of the book). Other pages composed by a single text block contain for instance short dedications. These layouts can be recognized with some features describing the position and size of the printed part of the page (represented in the MXY tree root) with respect to the document image. Heterogeneous collections contain different books with variable size. However, users are usually interested in pages with a given layout independently from the book size. In order to use features invariant with respect to the book size, the root position and the size of each page are normalized with respect to the bounding box of all the book pages. The computation of such a “book bounding box” is equivalent to the extraction of the bounding box of the image that is obtained by projecting all the book pages in a Z -direction (see Fig. 2). In so doing we are able to obtain features that are invariant with respect to different book sizes and different book locations in the scanner.

The features can be computed in the following way. Let (xb_i^j, yb_i^j) and (xe_i^j, ye_i^j) be the top-left and bottom-right points of the bounding box of page P_i in book B_j . The “book bounding box” can be simply computed by Eq. 1:

$$\begin{aligned}
 XB^j &= \min_{P_i \in B_j} (xb_i^j) & YB^j &= \min_{P_i \in B_j} (yb_i^j) \\
 XE^j &= \max_{P_i \in B_j} (xe_i^j) & YE^j &= \max_{P_i \in B_j} (ye_i^j)
 \end{aligned} \tag{1}$$

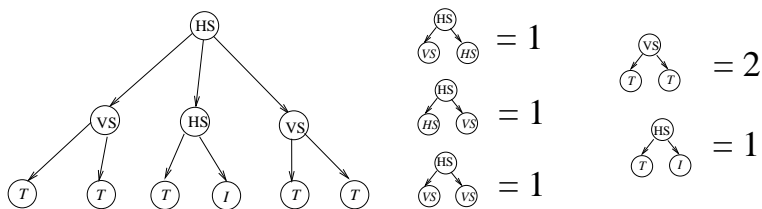


Fig. 3. A simple MXY tree; in the right part of the figure we show the balanced tree patterns in the tree, with the corresponding occurrences.

The page location can be described by computing the normalized position of the page center ($\bar{x}_i^j = \frac{xb_i^j + xe_i^j}{2}$, $\bar{y}_i^j = \frac{yb_i^j + ye_i^j}{2}$) with respect to the “Book Bounding Box” (Eq. 2):

$$x_i^j = \frac{\bar{x}_i^j - XB^j}{XE^j - XB^j} \quad y_i^j = \frac{\bar{y}_i^j - YB^j}{YE^j - YB^j} \quad (2)$$

The normalized width (w_i^j) and height (h_i^j) of page P_i in book B_j can be computed in the same fashion (Eq. 3):

$$w_i^j = \frac{xe_i^j - xb_i^j}{XE^j - XB^j} \quad h_i^j = \frac{ye_i^j - yb_i^j}{YE^j - YB^j} \quad (3)$$

2.2 MXY Tree Similarity

The layout similarity is computed on the basis of the similarity between MXY trees corresponding to the query and to the pages in the database. Although some limits of the XY decomposition have been pointed out in the literature, this representation is very appropriate when dealing with documents with a *Manhattan* layout and when the pages are not subjected to large skews. Digitized pages of both books and journals are examples of documents where these hypotheses are verified, and the XY tree representation is a good choice. In particular, MXY trees (that decompose the document image also along horizontal and vertical lines) have been demonstrated to be effective when dealing with documents containing ruling lines [9,10]. The MXY trees have been encoded into a fixed-size feature vector for page classification by taking into account occurrences of *tree-patterns* made by three nodes [9]. This approach is motivated by the observation that similar layouts frequently contain common sub-trees in the corresponding MXY tree. Trees composed by three nodes can have two basic structures: the first pattern has root and two children, whereas the second pattern (denoted as *balanced tree-pattern*, Fig. 3) is made by a root, a child, and a child of the second node.

MXY tree nodes contain symbolic attributes describing the purpose of the node. Basically, internal nodes represent the cut strategy considered (we can have horizontal/vertical cuts along either spaces or lines), whereas leaves correspond to homogeneous blocks in the page (text, image, horizontal/vertical line). Since node labels are in a fixed number, the number of possible *tree-patterns* (denoted

with TP) is fixed as well. Under these hypotheses, similar pages have some *tree-patterns* in common, and sometimes similar pages contain the same number of occurrences of a given *tree-pattern* (for instance table pages usually contain a large number of *tree-patterns* with lines). Unfortunately, there are some patterns that appear roughly in every document, and in this case these patterns are not very useful for measuring page similarities.

These peculiarities are very similar to the use of index terms in classic Information Retrieval. We extended the vector model approach, used in IR for dealing with textual documents, to our representation based on *tree-patterns*. The vector model of IR (see [1], Chapter 2) is based on a vectorial description of the document textual contents. The vector items are related to the occurrences of index terms, that usually correspond to words in the text of the document. Vector values are weighted in order to provide more importance to most discriminant terms. One common approach relies on the well known *tf-idf* weighting scheme. Basically, index terms that are present in many documents of the collection have a lower weight since their presence is not discriminant. In our approach, the vector model is used in order to describe the page layout. To this purpose, the page is described by means of the MXY tree representation, and occurrences of *tree-patterns* are used instead of word-based index terms. The extension of *tf-idf* weighting to this case is straightforward: the weight assigned to the k -th *tree-pattern* in the tree \mathcal{T}_j corresponding to page P_i is computed by Eq. 4.

$$w_{i,k} = f_{i,k} \cdot \log\left(\frac{N}{n_k}\right) \quad (4)$$

where $f_{i,k}$ is the frequency of the k -th *tree-pattern* in \mathcal{T}_j normalized with respect to the maximum *tree-pattern* frequency in \mathcal{T}_j , N is the total number of pages, and n_k is the number of trees containing the k -th *tree-pattern*.

2.3 Similarity Computation

The similarity between two pages is computed by taking into account the corresponding feature vectors that are made by two parts. The feature vector describing page P_i in book B_j can be represented as follows:

$$\boxed{\boxed{T} \mid \boxed{L} \mid \boxed{I} \mid \boxed{x_i^j} \mid \boxed{y_i^j} \mid \boxed{w_i^j} \mid \boxed{h_i^j} \mid \boxed{w_{i,1}} \mid \dots \mid \boxed{w_{i,k}} \mid \dots \mid \boxed{w_{i,TP}}}$$

The first seven values correspond to global features (Section 2.1). (T, L, I) are binary values describing the tree root. $(x_i^j, y_i^j, w_i^j, h_i^j)$ describe the position and size of the tree root (Eqs. 2 and 3). In a page containing a unique block the corresponding MXY tree is made by a unique node that can be a text block, a line, or an image block. These three cases are described with a mutual exclusion in T, L, I values (e.g. $T = 1, L = 0, I = 0$ corresponds to a text block). When the page contains more blocks, then the root does not correspond to a single block, and in this case the three values are all set to zero.

The rest of the vector contains an encoding of the MXY tree associated to the page (Section 2.2): $w_{i,k}$ $k = 1, \dots, TP$ are the weights associated to the occurrences of *tree-patterns* (Eq. 4).

The similarity between a query page q and a generic page p in the database is computed by combining two similarity measures for the two components of the feature vector. Let \mathcal{F} be the feature vector space, and $\mathbf{V} \in \mathcal{F}$ be a generic vector in \mathcal{F} . We indicate with \mathbf{V}_{GL} and \mathbf{V}_{XY} the two sub-vectors contained in \mathbf{V} . Let $\mathbf{Q} \in \mathcal{F}$ and $\mathbf{P} \in \mathcal{F}$ be the feature vectors corresponding to the query page q and to the page p , respectively. The similarity between q and p can be computed by Eq. 5

$$Sim(\mathbf{P}, \mathbf{Q}) = \alpha \cdot SimEuc(\mathbf{P}_{GL}, \mathbf{Q}_{GL}) + \beta \cdot SimCos(\mathbf{P}_{XY}, \mathbf{Q}_{XY}) \quad (5)$$

The similarity between \mathbf{P}_{GL} and \mathbf{Q}_{GL} is computed by using the Euclidean distance between the two sub-vectors (Eq. 6). The distance is divided by the maximum value that can be reached ($\sqrt{6}$). *SimEuc* has higher values (close to 1) when the pages are similar and the two sub-vectors are the same.

$$SimEuc(\mathbf{P}_{GL}, \mathbf{Q}_{GL}) = 1 - \frac{\sqrt{\sum_{i=1}^7 (P_{GL}[i] - Q_{GL}[i])^2}}{\sqrt{6}} \quad (6)$$

The similarity between \mathbf{P}_{XY} and \mathbf{Q}_{XY} is computed by taking into account the *cosine of the angle* between the two vectors (Eq. 7)

$$SimCos(\mathbf{P}_{XY}, \mathbf{Q}_{XY}) = \frac{\mathbf{P}_{XY} \times \mathbf{Q}_{XY}}{|\mathbf{P}_{XY}| \cdot |\mathbf{Q}_{XY}|} = \frac{\sum_{i=1}^{TP} (P_{XY}[i] \cdot Q_{XY}[i])}{\sqrt{\sum_{i=1}^{TP} P_{XY}[i]^2} \cdot \sqrt{\sum_{i=1}^{TP} Q_{XY}[i]^2}} \quad (7)$$

The two parameters α and β are used in order to weight the contribution of the two parts to the overall similarity measure. Several tests have been made by varying the values of α and β as it will be discussed in Section 4.

3 System Architecture

When designing a retrieval system that can easily scale up to large document collections, the use of a robust and reliable data-base management system (DBMS) is essential. The use of a standard DBMS is in contrast with approaches where all the information is kept in the file system. When dealing with large image collections one critical issue is related to image storage. One approach relies on the use of appropriate DBMSs that are able to store images. One intermediate strategy is based on the use of a DBMS for storing information about images, and to use customized approaches for image storage. In Digital Libraries, the image repositories are already defined and are usually based on complex organizations (based on standard file systems), where most difficulties are related to the use of appropriate hardware equipments (like juke-boxes for keeping large collections of CD-ROMS) [11].

To facilitate the integration with existing solutions we designed a system for computing the layout similarity, whereas document images are stored in the file system (this is clearly one of the main limits of the current implementation). We used the Java language for the user interface and the retrieval algorithms. The

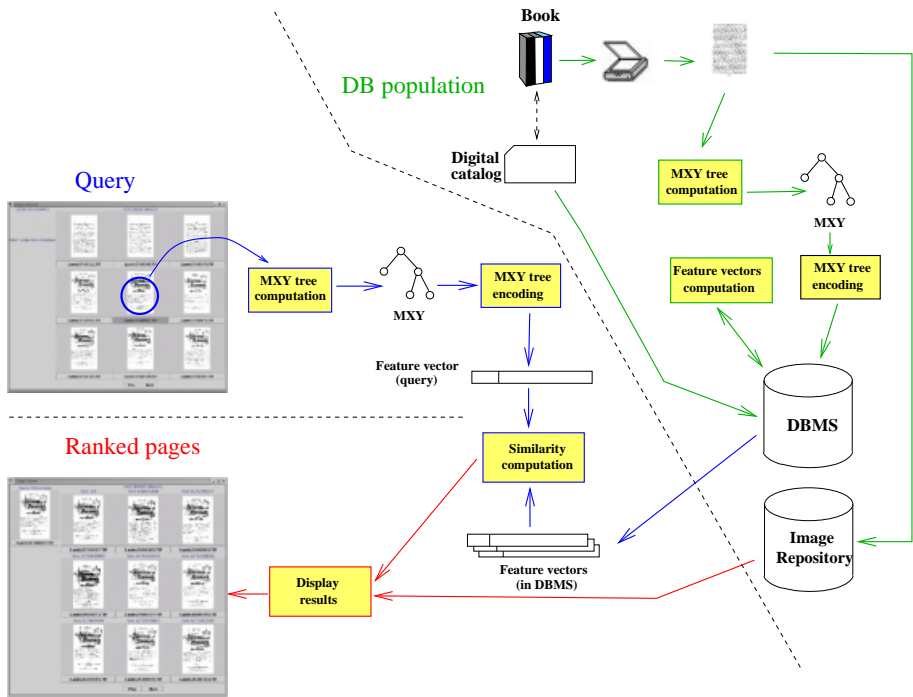


Fig. 4. System architecture.

DBMS currently used is IBM's DB2 [12], and the interface with the retrieval system is made through standard JDBC methods. The use of a DBMS for the storage of layout information allows us to obtain a dynamic system to which data (scanned books) can be easily added and removed. This is in contrast with static systems where the addition of new documents is a complex task. To support the dynamic updating of database contents, the MXY tree of each page is stored in the DBMS as well. The MXY trees are used at the end of an update session ("DB population" in Figure 4) in order to compute feature vectors on the basis of Eq. 4. When the user makes a query, the system computes the MXY tree of the sample page, and obtains the corresponding feature vector. According to Eq. 5 the pages are ranked by comparing the corresponding feature vectors with the vector computed for the sample page. Lastly, images of most relevant pages are retrieved from the image repository. The DBMS contains four tables: *Book*, *Page*, *XYNode*, and *Vector*. The *Book* table contains all the information useful for book identification. The main purpose of the table *Page* is to link together each page of a given book with the corresponding image file. The table *XYNode* is the largest table in the DBMS as it contains all the nodes of all the MXY trees in the database. Lastly, the *Vector* table contains the feature vectors corresponding to each document image.

4 Experimental Results

In this section we discuss the system evaluation that has been carried out using two sets of document images. The first set of images contains several issues of a journal, the second set contains a collection of books.

The evaluation of image retrieval systems is difficult for the peculiarities of the tasks required by users as well as for semantic ambiguities in defining image similarity [2]. In Information Retrieval, *Precision* and *Recall* are traditionally used for performance evaluation, and can be defined as follows (e.g. [1], page 75): $Precision = \frac{N_{RetRel}}{N_{Ret}}$, $Recall = \frac{N_{RetRel}}{N_{Rel}}$, where N_{Ret} is the number of retrieved documents, N_{Rel} is the number of relevant documents, and N_{RetRel} is the number of relevant documents in the set of retrieved documents.

There are two main limitations to the use of Precision and Recall in the field of image retrieval [2]. First, the selection of a relevant set in an image database is much more problematic than in a text database. Second, image databases usually do not return an undifferentiated set of “relevant” results, but a ranked list of all the documents in the collection sorted on the basis of their similarity with respect to the query. In document image retrieval the first problem is less critical than in more general CBIR systems, since frequently the relevance of a document with respect to a given query can be defined unambiguously. This assumption is true for the experiments reported in this paper, since all the pages in the collection belong to some user-defined classes, and consequently one page is clearly deemed to be relevant when its class corresponds to the query page class. However, the difficulty in computing Precision and Recall on the basis of a sorting of the complete collection of pages is an issue in our experiments as well. To provide an evaluation of the retrieval effectiveness, we introduce a measure (*accuracy*) that is appropriate in the problem at hand. Let N_{Ans} be the number of pages in the answer set. We can define the accuracy as follows:

$$Acc = \frac{N_{RetRel}}{\min(N_{Ans}, N_{Rel})} \tag{8}$$

This approach is appropriate when dealing with systems (like the one proposed in this paper) that provide to the user a ranked list of the N_{Ans} most relevant retrieved documents.






Table 1. Average accuracy on all the experiments when taking into account different values for α and β .

α	0	0.3	0.5	0.7	1
β	1	0.7	0.5	0.3	0
<i>Acc</i>	0.390	0.862	0.854	0.866	0.822

The aim of the tests described in this section is the evaluation of the system accuracy when different values of α and β are considered in Eq. 5. As illustrated in the following, the choice of these parameters is not too critical, provided that a contribution of both parts of the similarity measure are considered. The experiments have been made on two data sets that are representative of the material

stored in digital libraries. The first experiments have been made inserting in the system a collection of documents containing several issues of the journal “The American Missionary” belonging to the on-line digital library *Making of America*². In the second group of experiments we inserted in the database four scanned books of the 19th Century, containing 621 pages. These books have been downloaded from the on-line digital library hosted by the “*Bibliothèque Nationale de France*” (BNF)³. Examples of pages corresponding to each class are shown in Tables 2,3. For each test we used each page in the database as a query page, and we evaluated the accuracy on the basis of the classes of retrieved pages. Table 1 contains the average accuracy obtained for the whole set of documents when taking into account some values of α and β , and fixing $N_{Ans} = 10$ (similar results have been achieved with $N_{Ans} = 50$). More detailed results are shown in tables 2,3 whose numerical values correspond to the average accuracy computed for all the documents in a given class. Global accuracy values reported in table 1 give us two main messages. First, the best results are achieved when taking into account both similarity measures ($\alpha \neq 0$ and $\beta \neq 0$). In contrast, when considering only the MXY-tree based similarity ($\alpha = 0$) or the similarity relying on global features only ($\beta = 0$) lower values of *Acc* are reached. Second, the choice of appropriate values for α and β is not critical, since in a wide interval [$(\alpha = 0.3, \beta = 0.7)$, $(\alpha = 0.7, \beta = 0.3)$] the global accuracy is very similar.

Table 2. Average accuracy for pages in classes that are recognized better with the MXY tree based similarity.

Class (Book/Journal)	Advert (J)	Issue (B)	Sect0 (B)	Receipts (J)	Text2 (B)
$\alpha = 0, \beta = 1$	0.880	0.245	0.667	0.852	0.832
$\alpha = 1, \beta = 0$	0.700	0.204	0.667	0.604	0.768
$\alpha = 0.5, \beta = 0.5$	0.847	0.245	0.555	0.863	0.839
					
# pages	104	7	3	118	31
Class description	Two columns with pictures (advertisement)	Two columns with a centered title	Small text block	List of items on two columns	Two columns





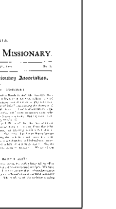


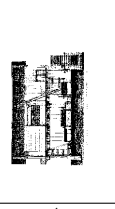

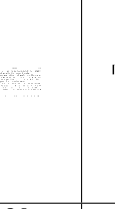


A detailed analysis of results obtained for each class is reported in Tables 2 and 3. In particular, Table 2 contains classes with a complex layout, and in this case it is quite natural that the results obtained considering only the MXY tree encoding are better than those obtained when considering global features only.

² Document images can be downloaded from the web site of the collection: <http://cdl.library.cornell.edu/moa/>

³ The web site of the collection is: <http://gallica.bnf.fr/>

One exception is class *Sect0* that presents a simple layout but is recognized with the same accuracy by both methods. Global features (Table 3) are appropriate when dealing with pages whose tree is made by a unique node (e.g. classes *Regular* and *Text1*). In this case the Accuracy is very low in columns where $\alpha = 0$, whereas higher values are obtained by setting $\alpha = 1$ and $\beta = 0$.

Table 3. Average accuracy for pages in classes that are recognized better with global features.

Class (Book/Journal)	Index (J)	Regular (J)	Text1 (B)	SecM1 (B)	First (J)
$\alpha = 0, \beta = 1$	0.895	0.081	0.064	0.015	0.876
$\alpha = 1, \beta = 0$	0.953	0.800	0.927	0.854	0.938
$\alpha = 0.5, \beta = 0.5$	0.907	0.791	0.925	0.407	0.942
					
# pages	107	172	478	26	107
Class description	Index with a large picture on top	Simple text on a single column	Simple text on a single column	Subsection title	First page of a chapter (one column)
Class (Book/Journal)	SecS1 (B)	Image (B)	ITLPA (B)	SecE1 (B)	Title (B)
$\alpha = 0, \beta = 1$	0.530	0.500	0.125	0.247	0.125
$\alpha = 1, \beta = 0$	0.770	0.750	0.313	0.296	0.312
$\alpha = 0.5, \beta = 0.5$	0.726	0.500	0.250	0.796	0.312
					
# pages	27	4	7	26	4
Class description	First page of a section	One large image	Table of contents	Last page of a section	Book title

5 Conclusions

We described a system for document image retrieval on the basis of layout similarity. Pages are represented with both global features and features related to the MXY tree layout representation. The similarity is computed by combining

the similarity measures that are defined for both types of features. In the case of global features we take into account a similarity that is based on the Euclidean distance between feature vectors. When dealing with MXY trees, the occurrences of some predefined *tree-patterns* are first counted in each tree. Afterwards, these occurrences are stored in a feature vector obtained with the *tf-idf* weighting scheme, and the similarity is evaluated by computing the *cosine of the angle* between the two vectors.

Future work concerns the implementation of a “query by sketch” approach, where the user can easily perform a query by drawing an ideal page sample. Moreover, we plan to add further retrieval strategies to this general framework by taking into account also information that can be extracted from the textual content of the page.

Acknowledgments. We would like to thank M. Ardinghi and S. Matucci for their work in the implementation of various parts of the system.

References

1. R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Addison Wesley, 1999.
2. A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, “Content-based image retrieval at the end of the early years,” in *IEEE Trans. PAMI*, vol. 22, pp. 1349–1380, December 2000.
3. D. Doermann, “The indexing and retrieval of document images: A survey,” *Computer Vision and Image Understanding*, vol. 70, pp. 287–298, June 1998.
4. M. Mitra and B. B. Chaudhuri, “Information retrieval from documents: a survey,” *Information Retrieval*, vol. 2, pp. 141–163, 2000.
5. D. Doermann, J. Sauvola, H. Kauniskangas, C. Shin, M. Pietikainen, and A. Rosenfeld, “The development of a general framework for intelligent document image retrieval,” in *Document Analysis Systems*, pp. 605–632, 1996.
6. C. Shin and D. Doermann, “Classification of document page images based on visual similarity of layout structures,” in *SPIE 2000*, pp. 182–190, 2000.
7. J. Hu, R. Kashi, and G. Wilfong, “Comparison and classification of documents based on layout similarity,” *Information Retrieval*, vol. 2, pp. 227–243, May 2000.
8. J. F. Cullen, J. J. Hull, and P. E. Hart, “Document image database retrieval and browsing using texture analysis,” in *Proceedings of the 4th International Conference on Document Analysis and Recognition*, pp. 718–721, 1997.
9. F. Cesarini, M. Lastrì, S. Marinai, and G. Soda, “Encoding of modified X-Y trees for document classification,” in *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, pp. 1131–1136, 2001.
10. F. Cesarini, M. Gori, S. Marinai, and G. Soda, “Structured document segmentation and representation by the modified X-Y tree,” in *Proc. of ICDAR '99*, pp. 563–566, 1999.
11. W. Y. Arms, *Digital Libraries*. MIT Press, 2000.
12. G. Baklarz and B. Wong, *DB2 Universal Database V7.1*. Prentice Hall, 2000.