2012 10th IAPR International Workshop on Document Analysis Systems

An efficient coarse-to-fine indexing technique for fast text retrieval in historical documents

Partha Pratim Roy, Frédéric Rayar, Jean-Yves Ramel
Laboratoire d'Informatique
Université François Rabelais
Tours, France
{*partha.roy, frederic.rayar, ramel*}*@univ-tours.fr*

***Abstract*—In this paper, we present a fast text retrieval system to index and browse degraded historical documents. The indexing and retrieval strategy is designed in a two level, coarse-to-fine approach, to increase the speed of the retrieval process. During the indexing step, the text parts in the images are encoded into sequences of primitives, obtained from two different codebooks: a coarse one corresponding to connected components and a fine one corresponding to glyph primitives. A glyph consists of a single character or a part of a character according to the shape complexity. During the querying step, the coarse and the fine signature are generated from the query image using both codebooks. Then, a bi-level approximate string matching algorithm is applied to find similar words; using coarse approach first, and then the fine approach if necessary, by exploiting predetermined hypothetical locations. An experimental evaluation on datasets of real life document images, gathered from historical books of different scripts, demonstrated the speed improvement and good accuracy in presence of degradation.**

***Keywords*-Word Spotting, Historical Documents, Approximate String Matching;**

I. Introduction

Text searching in historical document is getting popular in Document Image Processing (DIA) research community due to its complexity and the growing necessity for accessing the content of the book. In recent years, mass digitization of historical documents in libraries, museums have been done and these digital information are made available to users through web-portals. By these portals, users are restricted to access only to view the pages (already scanned). Searching with content information (e.g. word) is available if the books are transcribed. Due to volume of data, manual transcription is not feasible. However, automatic text transcription, performed by the available commercial OCR systems are not satisfactory. OCR provides poor performance of transcription in such images. The difficulty arises from the severely touching or broken characters due to degradation occurred by ageing, strains, repetitive use, etc.

Text searching using word spotting techniques [1] provides an alternative approach for indexing and retrieval. It treats each word as a whole entity and thus avoids the difficulty of character segmentation and recognition. Word spotting produces a ranked list of word images according to similarity of the query word image. The matching is done at image level through word shape coding, produced by a set of features at different zones of the word image. Rath and Manmatha [1] applied dynamic time warping (DTW) distance to a set of features for matching similar words. However these approaches rely on accurate segmentation of words [2]. If the words are over-segmented or under-segmented due to noise from background, these approaches may fail.

To take care of the word segmentation problem, researchers try to solve this problem using segmentation free approaches. For this purpose, Leydier et al. [3] have used differential features and a cohesive elastic distance. Gatos and Pratikakis [4] proposed an approach using salient region detection by template matching at an initial stage. Recently, Hidden Markov Model (HMM) based method [5] has been applied in text lines for finding the query text word written by different writer. Rusiñol et al. [6] used bag of visual words generated by densed SIFT features to detect the query keyword without segmentation of the page.

However, most of these methods are of huge complexity and time consuming because they need to analyze in detail and compute complex features for all parts of the images during the indexation step. We show a portion of sample document image from our dataset in Fig.1. There exist some layout segmentation and text line segmentation tools (e.g. AGORA [7]) that can take care of the complex layout in noisy historical document images. But, often due to non-uniform inter-character spacing and degradation, proper word segmentation is difficult to obtain.

In normal printed documents, shape coding is often used to encode the words [8]. Inspired with this idea, we proposed an approach for word retrieval in collection of historical documents using primitive segmentation in [9]. The present work extends the work by reformulating the strategy of indexation of noisy text documents into a two level coarse to fine approach to speed up the retrieval process. During the indexing step, only the text parts in the images are encoded in sequences of primitives taken from two different codebooks: a coarse one corresponding to connected components and a fine one corresponding to glyph primitives. A

978-0-7695-4661-2/12 $26.00 © 2012 IEEE
DOI 10.1109/DAS.2012.17

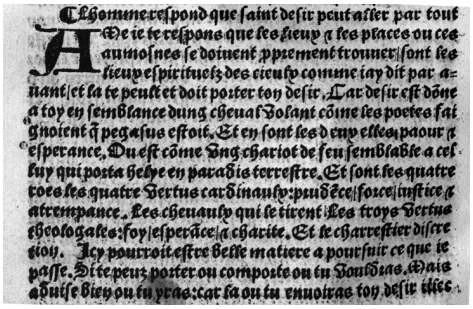


Figure 1. A portion of historical document from the collection shows (a) difficulty to group characters in a word due of non-uniform spacing, (b) character recognition problem due to strains and degradation.

glyph consists of a single character or a part of a character according to the shape complexity. During the querying step, the coarse and the fine signatures are generated form the query image using both the codebooks. Next, a bi-level approximate string matching algorithm is applied to find similar words; using coarse approach first; and fine approach from the predetermined hypothetical locations only if necessary.

The proposed approach can be used in different scripts as the method uses dynamic codebook vocabulary for text encoding. The main contributions of this paper are the use of text portions (primitives) instead of whole word and encoding the text using these primitives to generate coarse and fine level of indexing. The advantage of this approach is that it searches for possible words in efficient way using coarse level of primitive shapes (i.e. connected component) first. Then, if necessary, it uses fine primitives (i.e. glyph component) to detect strings of touching and broken characters. Our approach does not need proper segmentation of words and we use text lines as input for word spotting; thus it avoids the word segmentation problem. As the method searches for query word using string matching in terms of primitives, it is fast and tolerant to noises.

The rest of the paper is organized as follows. In Section 2, we explain in detail the proposed indexing approach. In section 3, we discuss the word retrieval process when the query word is provided. Section 4 presents the experimental results in datasets of different scripts. Finally conclusion is given in Section 5.

II. Text Indexing Approach

Layout segmentation and text line segmentation of document images are pre-requisite in our text indexing scheme. In our system, we have used AGORA layout analysis tool [7] for text blocks segmentation because of its superior performance in historical document analysis. The system works with interactive scenario analysis from user feedback. The user builds scenarios according to his needs (location of the dropcaps, notes at margins, etc.) allowing labelling, merging or removing blocks contained in the intermediate representation. These scenarios are next applied to the rest of the images in batch processing. Finally, AGORA provides segmentation and labelling of each block (text line, picture, dropcaps, etc.) in the document. Next each of these text lines are indexed using our approach as discussed below.

A. Primitive Selection

Our text indexing approach is based on the primitives extracted from the text lines and coding them using primitive vocabulary. The idea is, character objects can be represented by a small set of shared primitives. We use two different primitives in our approach: connected component (CC) and glyph. A connected component describes the pixels connected together. A glyph consists of a single CC (character) or a part of a CC according to the shape complexity. In the following sections, we use the term "primitive" to describe in general CC and glyph.

To obtain the primitives, we first apply a connected component analysis to the text line image and extract individual components. Each CCs are next segmented into glyphs according to its background information obtained by water reservoir concept [10]. Using this concept, if water is poured from a side of a component, the cavity regions of the background portion of the component where water will be stored are considered as reservoirs of the component. Since, touching of neighbor characters affects mostly the upward or downward reservoir rather leftward or rightward reservoirs, we have used only the information of top and bottom reservoirs. Next, the selected top (bottom) reservoirs are segmented at lowest (highest) reservoir point and component is split into glyph primitives.

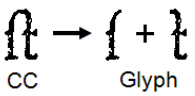


Figure 2. Different primitives: CC and Glyph.

As our method segments primitives from connected components, the text characters (e.g. 'i', '$\grave{a}$', '$\acute{e}$', etc.) having multiple components will also be segmented into different primitives. These isolated components are grouped into a single primitive by checking their overlapping positions [9].

B. Codebook of Primitives

We have generated two different codebooks: a coarse one corresponding to CC and a fine one corresponding to glyph. For each codebook, the representative primitives are learnt through an unsupervised clustering of corresponding primitives extracted in training. The clustering is performed in an incremental fashion. The similarity (S) between two primitives (A and B) is measured by the template matching using cross correlation equation given by:

$$S(A,B) = \frac{\sum_{y=0}^{h-1}\sum_{x=0}^{w-1}\tilde{A}(x,y)\tilde{B}(x,y)}{\sqrt{\sum_{y=0}^{h-1}\sum_{x=0}^{w-1}\tilde{A}(x,y)^2\sum_{y=0}^{h-1}\sum_{x=0}^{w-1}\tilde{B}(x,y)^2}}$$

where, $\tilde{A}(x,y) = A(x,y) - \bar{A}$ and $\tilde{B}(x,y) = B(x,y) - \bar{B}$. $\bar{A}$, $\bar{B}$ are the mean of pixel values. h and w are the normalized height and width of image size. The codebooks are generated from a set (20%) of training pages of the book. These ensure that most of the basic primitives are present in the codebook.

C. Indexing by Codebook

After the codebooks are created from the segmented primitives of the training pages, each text lines from the book are indexed in two different passes using the codebook primitives. To generate the index files, the codebook primitives are first classified and indexed by unique labels $L^m = \{L_1, L_2, \ldots, L_m\}$. Where, m is the total number of primitives in the codebook. Next, the segmented primitives of the text line are tagged using the label of most similar codebook primitive. Due to noise, a primitive may not be represented by a same codebook primitive always. To take care of this problem, during index generation we record more than one nearest codebook primitives for each primitive. The ranking of nearest primitives are done by similarity measure (S). In our experiment, we have used 3 nearest codebook primitive models. Fig.3 details the general indexing scheme used in our approach. We generate index files for both the primitives: CC and glyph.

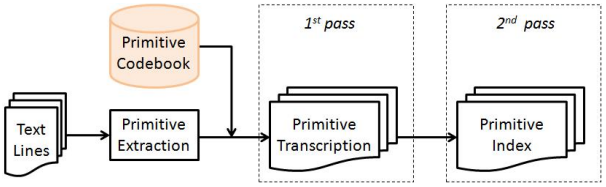


Figure 3. Steps of the text indexing approach.

In the first pass, we generate the transcription of text lines with the representatives of codebook. Each text line (T) is translated into two sequences of primitive labels. Thus, the word image of two dimension has been converted to 2 one dimensional strings of codebook labels of L^m (CC based string and glyph based string). We keep the positions of the primitives along with the string in the file for retrieval purpose. To make the retrieval process faster, we have generated a second index file. This file compiles all the positions of primitives and organizes them according to the primitive. Finally each of the primitives store the occurrences of all the text lines in the book. These information allow us to find the corresponding text lines in fast way.

III. Retrieval of Query Word

For searching a query word Q from the collection of text lines, we can use either a single primitive based approach (CC or glyph) or a combination of CC and glyph based approach. Both these approaches are described below. In both of these approaches we use string matching algorithm. Approximate String Matching (ASM) algorithm [11] has been used in our system for text searching. The algorithm finds all substrings of the text T that have at most k errors (character that are not same) with the query pattern Q. When $k = 0$ (no mismatches) it is simple string matching algorithm. The ASM is adapted to handle more than one choices of each primitive [9].

A. Single Primitive based Retrieval

We use text retrieval using single primitive (either CC or glyph) based searching. To do so, Q is segmented into corresponding primitives and then encoded into a sequence of primitive labels as explained in Section 2. Now, the matching between query word Q and a target word T is formulated as matching of 2 sequences of primitives. To faster the retrieval process, the text lines are filtered first. We use the second pass indexing files and find the candidate text lines using primitive obtained in query text. Next, these candidate text lines are fed to the ASM algorithm.

B. Retrieval using Combination of Primitives

To improve the overall performance of text retrieval, we have used a combination of coarse-to-fine primitive matching. For this purpose, we generate the coarse as well as fine primitive signatures (if necessary) of query word. We start with CC based primitive searching. The candidate lines are matched using CC based ASM. If the result is not satisfactory (matching score more than T_{d1}), we look for the glyph based signature and find the results using glyph based ASM. In Fig.4, we show the flowchart of our retrieval approach. The string distance threshold T_{d1} and T_{d2} are set up according to experimental results. We fixed them as 1 + T_{len}/4, where T_{len} is the length of string.

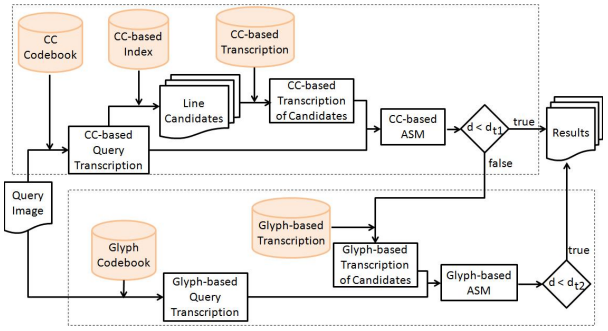


Figure 4. Flowchart of query retrieval using combination of index files.

IV. Experimental Setup and Results

The "Centre d'Études Supérieures de la Renaissance" (CESR) of Tours has created in 2002 the Humanistic Virtual Library [12], that contains bitmap versions of several books. They have a collection of precious historical books, currently numbering around 3000 copies dating from the middle of the XIV century to the beginning of the XVII century. Some of them are already scanned or photographed, and made

available. Latin and French are the most frequent languages used in these books. An example of scanned page of these books is shown in Fig.1.

Experiments have been performed to test the retrieval of words from two different books of this collection: *"Essais - Livre I"* of Montaigne and *"La mendicite spirituelle"*. A ranking based on string matching distance is done to evaluate the retrieved word. When two or more words are found with similar distance, they are discriminated by the accumulated similarity measures of primitives. The results will be presented as follow: the query image is shown in the first row, then three results are displayed to get an idea of the qualitative performance.

In Fig.5, the results are obtained for queries with different kinds of known degradation issues in historical documents, such as spilled ink, touching or broken characters. The use of glyph primitive and a coarse-to-fine retrieval approach, allow the system to be robust against these issues.

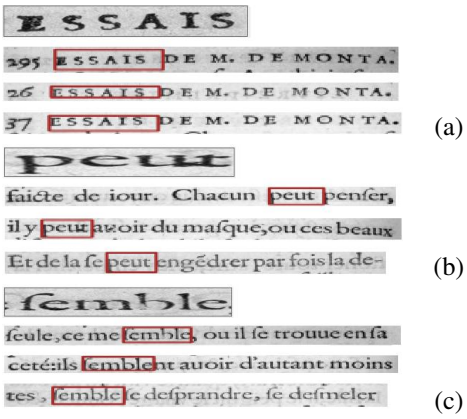


Figure 5. Results obtained for queries in book *"Essais - Livre I"* showing effectiveness to handle (a) spilled ink, (b) touching characters, (c) broken characters.

Even if the text in *"Essais - Livre I"* of Montaigne is affected by various degradation, an acceptable (not perfect) word segmentation method could be considered to provide separate words. This is not the case for all the historical books. Fig.6 presents the retrieval results in book *"La mendicite spirituelle"*, where inter-word space is not even, nor relevant enough to perform an exact word segmentation. Thus, with our approach, line segmentation of pages is a sufficient granularity.

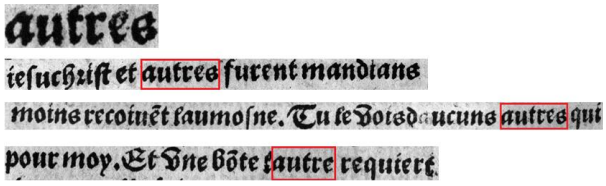


Figure 6. Retrieval in a book where word segmentation may not work.

The proposed method is script independent, thanks to the creation of dynamic codebook vocabulary during the indexing process. We show some results in Fig.7 obtained in datasets of Hindi and Bengali scripts. These images are collected from newspapers and scanned in 200 dpi. We used our system for searching the query words without changing the parameters and obtained good results. It is to be noted that, our approach can take care of extra character shape in the middle of a target word during retrieval (See 3rd result of Fig.7(b)). This is very effective for such scripts.

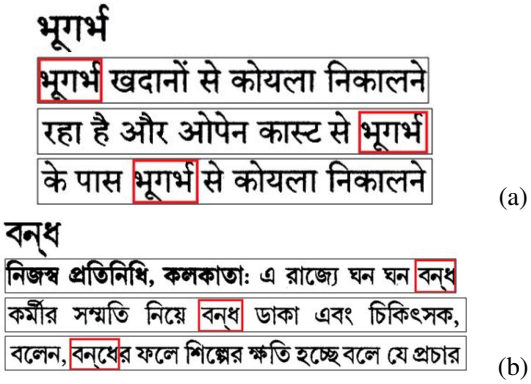


Figure 7. Results obtained for (a) hindi and (b) bengali script documents.

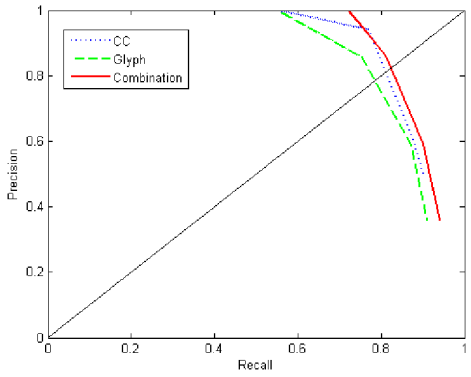
Quantitative performance of the system has been evaluated on *"Essais - Livre I"* dataset. This choice has been motivated by the fact that ground truth transcription has been done by the CESR and made available to measure the performance of our retrieval system. This dataset is composed of 78 pages from the book. We used AGORA to extract the text lines and made a pruning to remove irrelevant lines (*e.g.* isolated characters). Finally, 1579 lines have been considered for the experiments.

Common ratio of precision (P) and recall (R) have been used as performance measures. Each retrieved word is considered as relevant or not depending on the ground truth of the data. 5 query word images have been used and submitted to 3 different approaches: (a) CC-based approach by considering only the CC information, (b) glyph-based approach by considering only the glyph information and (c) using the proposed coarse-to-fine approach, by considering a combination of CC and glyph information. Fig.8 presents the 5 queries and the average P-R curve obtained from them.

We can notice that the coarse-to-fine approach outperforms both the CC-based approach and glyph-based approach. Up to 72% of the relevant results are found with a 100% precision by using the combination of CC and glyph. It is to be noticed that the CC-based approach has better performance than the glyph-based approach in this dataset. It is because, using glyph, we sometimes obtain over segmentation of components which affects the matching. The combination approach takes the benefit of both CC and Glyph based methods. It ranks the results according to the distance found from CC and Glyph based method and hence provides better precision result.

One of the contribution of the proposed indexing approach

(a)



(b)

Figure 8. Average P-R curve in (b) has been generated with 5 different queries shown in (a).

is the speed improvement of the online retrieval of query text. We have used a PC of "I5 CPU of 2.53Ghz and 4G RAM" to test our methods. In Table I we show the time taken using queries of various length: (a) small (< 5 letters), (b) medium size (5-10 letters) and (c) large size (>10 letters or multiple words). For each query, average runtime has been computed from different runs made in the experiment. Results show an average time of 1 to 2 seconds for a word or a text query. We measured the time of our previous approach [9] which was approximately 5 seconds in average. Optimization and parallelization of the code are yet to be done, which promises better performances in terms of speed improvement.

Table I
RUNTIME OF THE PROPOSED APPROACH USING COMBINATION

Query Word	Time (in ms)
(a) roy	1237
(b) Marefchal	2514
(c) ESSAIS DE M. DE MONTA.	4537

V. CONCLUSION

We have presented a robust and fast word spotting system for historical documents. A two level approach in terms of coarse-to-fine is proposed to increase the speed of the retrieval process. We use connected components as well as glyph primitives for the indexing purpose. During the querying step, the primitives search the possible locations of the primitives using indexed location. Finally, a string matching algorithm is used to retrieve the similar words from the collection.

The proposed approach works on segmented lines and thus avoids the word segmentation problem in noisy documents. As the codebook vocabulary is based on coarse and fine primitives, efficient retrieval is performed to take care of touching/broken character problem. The methodology has been made generic and tested in different scripts. In future we want to extend this system for OCR or transcription purpose.

VI. ACKNOWLEDGEMENTS

This work has been supported by the AAP program of Université François Rabelais, Tours, France (2010-2011) and by the Google Digital Humanities Research Awards (2000) given to the Computer Science Laboratory of Tours (RFAI team).

REFERENCES

[1] T. M. Rath and R. Manmatha, "Word image matching using dynamic time warping," in *Proceedings of CVPR*, vol. 2, 2003, pp. 521–527.

[2] R. F. Moghaddam and M. Cheriet, "Application of multi-level classifiers and clustering for automatic word spotting in historical document images," in *Proceedings of ICDAR*, 2009, pp. 511–515.

[3] Y. Leydier, A. Oujia, F. LeBourgeois, and H. Emptoz, "Towards an omnilingual word retrieval system for ancient manuscripts," *Pattern Recognition*, vol. 42, pp. 2089–2105, 2009.

[4] B. Gatos and I. Pratikakis, "Segmentation-free word spotting in historical printed documents," in *Proceedings of ICDAR*, 2009, pp. 271–275.

[5] A. Fischer, A. Keller, V. Frinken, and H. Bunke, "Hmm-based word spotting in handwritten documents using subword models," in *Proceedings of ICPR*, 2010, pp. 3416–3419.

[6] M. Rusiñol, D. Aldavert, R. Toledo, and J. Lladós, "Browsing heterogeneous document collections by a segmentation-free word spotting method." in *Proceedings of ICDAR*, 2011, pp. 63–67.

[7] J.-Y. Ramel, S. Leriche, M. L. Demonet, and S. Busson, "User-driven page layout analysis of historical printed books," *IJDAR*, vol. 9, no. 2-4, pp. 243–261, 2007.

[8] S. Lu, L. Linlin, and C. L. Tan, "Document image retrieval through word shape coding," *PAMI*, vol. 30, pp. 1913–1918, 2008.

[9] P. P. Roy, J. Ramel, and N. Ragot, "Word retrieval in historical document using character-primitives," in *In Proceedings of ICDAR*, 2011, pp. 678–682.

[10] U. Pal, A. Belaid, and C. Choisy, "Touching numeral segmentation using water reservoir concept," *Pattern Recognition Letters*, pp. 261–272, 2003.

[11] P. A. V. Hall and G. R. Dowling, "Approximate string matching," *ACM Computing Surveys*, vol. 12, pp. 381–402, December 1980.

[12] CESR, "Les bibliothèques virtuelles humanistes," http://www.bvh.univ-tours.fr/index.htm.