A Document Image Retrieval System

Konstantinos Zagoris, Kavallieratou Ergina, Nikos Papamarkos

Abstract

In this paper, a system is presented that locates words in document image archives. This technique performs the word matching directly in the document images bypassing character recognition and using word images as queries. First, it makes use of document image processing techniques, in order to extract powerful features for the description of the word images. The features used for the comparison are capable of capturing the general shape of the query, and escape details due to noise or different fonts. In order to demonstrate the effectiveness of our system, we used a collection of noisy documents and we compared our results with those of a commercial optical character recognition (OCR) package.

Keywords: Document retrieval, Word spotting, Segmentation, Information retrieval, Feature extraction

1. Introduction

In the last years, the world has experienced a phenomenal growth of the size of multimedia data and especially document images, which have been increased thanks to the ease to create such images using scanners or digital cameras. Thus, huge quantities of document images are created and stored in image archives without having any indexing information. In order to satisfactorily exploit these collections of document images, it is necessary to develop effective techniques to retrieve the document images.

A detailed survey on document image retrieval up to 1997 can be found in Doermann (1998). Historically, the use of index of descriptors for each document provided manually by experts was the first approach to the problem (Salton, 1989).

Next, with the improvement in character recognition field, optical character recognition (OCR) packages were applied to documents in order to convert them to text. These techniques transformed the characters, which were contained in the image into a machine-editable text. Thus, Edwards (2004) described an approach to transcribing and retrieving Medieval Latin manuscripts with generalized Hidden Markov Models. Their hidden states correspond to characters and the space between them. The training instance is used per character and character n-grams are used, yielding a transcription accuracy of 75%. Tan et al. (2002) described an approach to retrieve machine printed documents with a textual query, not necessarily in ASCII notation.

He describes both the query and the words occurring in the document images with features, which may then be matched in order to identify query term occurrences. A disadvantage of the above approaches is the considerably low noise tolerance, which yields low retrieval scores (Ishitani, 2001).

More recently, with the improvement in document image processing (DIP) field, techniques that make use of images instead of OCR were also introduced. Leydier et al. (2005) used DIP techniques to create a pattern dictionary of each document and then they performed word spotting by selecting the feature of the gradient angle and a matching algorithm. Kolcz et al. (2000) described an approach for retrieving handwritten documents using word image templates. Their word image comparison algorithm is based on matching the provided templates to segmented manuscript lines from the Archive of the Indies collection. Konidaris et al. (2007) proposes a technique for keyword guided word spotting in historical printed documents. He creates synthetic image words as query and performs word segmentation using dynamic parameters and hybrid feature extraction. Finally, he uses user feedback to optimize the retrieval. Matching of entire words in printed documents is also performed by Balasubramanian et al. (2006). In this approach, a dynamic time warping (DTW) based partial matching scheme is used to overcome the morphological differences between the words. Similar technique is used in the case of historical documents (Rath and Manmatha, 2003) where noisy handwritten document images are preprocessed into one-dimensional feature sets and compared using the DTW algorithm. Rath et al. (2004) presented a method for retrieving large collections of handwritten historical documents using statistical models. Using a word image matching algorithm, he clustered occurrences of the
same word in a collection of handwritten documents. When clusters that contain index terms are labeled, a partial index can be built for the document corpus, which can then be used for ASCII querying. The limitation of the above approach is the language dependent statistical model it requires.

Lu and Tan (2004) presented an approach with the capability of searching a word portion in document images. A feature string is synthesized according to the character sequence in the user-specified word, and each word image extracted from documents is represented by the corresponding feature string. Then, an inexact string matching technology is utilized to measure the similarity between the two feature strings. Unfortunately the above procedure has the same problems with noisy documents although it bypasses the OCR technique.

In this paper, we propose a Document Image Retrieval System (DIRS) based on word spotting, which has a high noise tolerance and is language independent. The proposed technique encounters the document retrieval problem using a word matching procedure, which performs the word matching directly in the document images bypassing OCR and using word images as queries. The entire system consists of the offline and the online procedures. In the offline procedure, the document images are analyzed in order to locate the word limits inside them. Then a set of features capable of capturing the word shape and discard detailed differences due to noise or font differences are calculated from these words and the results are stored in a database. The user, in the online procedure, enters a query word and then the proposed system creates an image of it and extracts the same set of features. Consequently, these features are used in order to find similar words through a matching procedure. Finally, the documents that contain these similar words are presented to the user. Experimental results show that the proposed document retrieval system has high mean precision and mean recall rates when it is applied on a collection of noisy documents. Also, comparative results with a commercial OCR package gave better results while experiments for different sizes and styles of fonts did not produce significant change to the performance.

The rest of this paper is organized as follows. Section 2 describes the overall structure framework of the system while Section 3 presents and analyzes the features that are used. The matching procedure of our system is described in detail in Section 4. Sections 5 and 6 describe the implementation of the proposed system and present some test results, respectively. Finally, in Section 7 we draw some conclusions and the future directions of our research.

2. The Document Image Retrieval System (DIRS)

The overall structure of the proposed system is presented in Fig. 1. It consisted of two different parts: the offline and the online procedure. An analytical description of the different stages of both procedures is as follows.

2.1. Preprocessing stage

In the offline operation, the document images are analyzed in order to localize the word limits and the results are stored in a database. This procedure consists of three main stages. Initially, the document images pass the preprocessing stage, which consists of a median filter (Fig. 2(b)), in order to face the existence of noise, e.g. in case of historical or badly maintained documents, and a binarization method (Fig. 2(c)). The median filtering is a nonlinear signal processing technique developed by Tukey (1974) that is useful for noise suppression in images (Pratt, 2007). The binarization is achieved by using the well-known Otsu technique (Otsu, 1979). This technique performs binarization through the histogram of the image by minimizing the inter-class variance between background and foreground pixels.

2.2. Word segmentation

The word segmentation stage follows the preprocessing stage. Its primary goal is to detect the word limits and filter the noise and punctuation marks. This is accomplished by using the connected components labeling and filtering method.

Firstly, all the connected components (CCs) (Fig. 3(a)) are identified. Since the noise of some documents can change significantly the shape of the extracted word images, the noise and the punctuation points must be rejected. To achieve this, the
most common height of the CCs (CCch) is calculated and those with height less than 70% of the CCch are rejected. In Kavallieratou et al. (2002), it has been proven that the height of a word can reach the double of a character mean size due to the presence of ascenders and descenders. This means that the applied CCs filtering can only reject areas of punctuation points, noise, etc. (Fig. 3(b)).

Due to the facts that Kavallieratou et al. (2002) present about the mean character size and having filtering out accents, noise and punctuation marks, it is rather rare of the same word to have distance greater than 20% of the CCch and different words to be closer than that. In order to merge the CCs and create the word blocks, the left and right sides of the CCs are expanded (Fig. 3(b)) by 20% of the CCch as Fig. 3(c) depicts. Finally, to locate the words, the overlapping CCs are merged (Fig. 3(d)).

3. Features

The proposed system is based on six features that are extracted from every word capable of capturing the word similarities and discarding the small differences due to remaining noise or different style of fonts. They are carefully selected in order to describe the contour and region shape of the word.

The six-feature-set is

1. **Width to height ratio:** The width to height ratio of the word forms important information concerning the word shape.

2. **Word area density:** This feature represents the percentage of the black pixels included in the word bounding box. It is
calculated by using the following relation:

\[
E = 100 \left( \frac{BP}{IW \cdot IH} \right)
\]

where \((BP)\) is the number of black pixels in the word bounding box, \((IW)\) is the width and \((IH)\) is the height of the word bounding box in pixels.

(3) Center of gravity: It represents the Euclidean distance from the word’s center of gravity to the upper left corner of the bounding box. In order to calculate this, the vertical and horizontal center of gravity must be determined by the following equations:

\[
C_x = \frac{M_{1,0}}{M_{0,0}}
\]

\[
C_y = \frac{M_{0,1}}{M_{0,0}}
\]

where \(C_x\) is the horizontal center and \(C_y\) the vertical center of gravity and \(M_{p,q}\) the geometrical moments of rank \(p+q\):

\[
M_{p,q} = \sum_x \sum_y \left( \frac{x}{\text{width}} \right)^p \left( \frac{y}{\text{height}} \right)^q f(x,y)
\]

The \(x\) and \(y\) determine the image pixels. The image is binary, so the \(f(x,y)\) is considered to be 1 when the pixel \((x,y)\) is black and 0 when the pixel is white. The division of \(x\) and \(y\) by the width and the height of the image, respectively, causes the geometrical moments to be normalized and be invariant of the size of the word. Finally, the center of the gravity feature is defined as equal to the Euclidean distance from the upper left corner of the image:

\[
\text{COG} = \sqrt{C_x^2 + C_y^2}
\]

(4) Vertical projection: This feature consists of a vector with twenty (20) elements, extracted from the smoothed and normalized vertical projection of the word image (Fig. 4). These elements correspond to the first twenty (20) coefficients of the discrete cosine transform (DCT) of the smoothed and normalized vertical projection. The smooth and normalized vertical projection has common width and height properties for all the word images and its calculation consists of the following steps:

Step 1: The vertical projection \(VP_{orig}[i]\) (Fig. 4(b)) is defined as the number of black pixels that they are residing in each column \(i\).

Step 2: From the Eq. (6) a new projection \(VP[i]\) is produced, which has width \(l\) and maximum height \(h\). In our proposed method, \(l\) and \(h\) are equal to the mean width and height, respectively, of all the word bounding boxes. For our experiment database \(l = 167\) and \(h = 50\):

\[
VP[i] = \frac{h}{h_{\text{max}}} \left( \frac{VP_{orig}[i]}{l_{\text{orig}}/l} \right)
\]

\(l_{\text{orig}}\) is the original width of the original projection \(VP_{orig}[i]\) and \(h_{\text{max}}\) the height of the word bounding box.

Step 3: The final normalized vertical projection depicted in Fig. 4(c) is created after applying a \(5 \times 1\) mean mask to the projection \(VP[i]\). This way, the final projection is more robust to the changes of the size and type of fonts.

(5) Top–bottom shape projections: As shown in Fig. 5, the top–bottom shape projections can be considered as signatures of the word shape. These signatures lead to a feature vector of 50 elements, where the first 25 values are the first 25 coefficients of the smoothed and normalized top shape projection DCT (Fig. 5(d)) and the rest 25 values are equal to the first 25 coefficients of the smoothed and normalized bottom shape projection DCT (Fig. 5(e)).

In order to calculate the top shape projection, the word image is scanned from top to bottom. As shown in Fig. 5(b), the first time a black pixel is found all the following pixels of the same column are converted to black.

The bottom shape projection is found similarly. As shown in Fig. 5(c), the word image is scanned from bottom to top and all the pixels are converted to black until a black pixel is found. The smoothed and normalized shape projections (Fig. 5(d) and (e)) are calculated as described at the vertical projection paragraph.

(6) Upper grid features: The upper grid features (UGF) is a ten element vector with binary values extracted from the upper part of each word image. In order to calculate this, initially the image’s horizontal projection is extracted, and from it, the upper part of the word is determined by the following algorithm:

Step 1: Smooth the horizontal projection with a mean \(5 \times 1\) mask.

Step 2: Starting from the top, find the position \(i\) in the horizontal projection histogram \(V(i)\) where \(V(i) \geq H\) as depicted in Fig. 6(a). The \(H\) is the maximum height of the horizontal projection (max\(V(i)\)). If the position \(i\) is below the half of the world then the word has no upper part.

---

**Fig. 4.** A visual representation of the vertical projection calculation: (a) original image; (b) the vertical projection of the original image; and (c) the smoothed and normalized vertical projection.

**Fig. 5.** A visual representation of the top–bottom shape projection calculations: (a) original image; (b) original top shape projection; (c) original bottom shape projection; and (d) the smoothed and normalized top shape projection and (e) the smoothed and normalized bottom shape projection.
Step 3: Find the position \( k_A(0, i) \) in the \( V(i) \) horizontal projection histogram when \( V(k) - V(k - 1) < 0 \). Then \( k \) defines the upper part of the word. If \( k \) has a very small value (3 or 2) the word has no upper part.

Then the upper part of the word is separated in ten similar parts as depicted in Fig. 6(b). The number of black pixels is counted for each part. If this number is bigger than the height of the extracted image, the relative value of the vector is set to 1; otherwise it is set to 0. Fig. 6(b) and (c) illustrates an example. The height of the extracted image is 43. The obtained feature vector is shown in Fig. 6(d).

**Down grid features:** As the name suggests, they are similar to the UGF but they are extracted from the lower part of the word image. The down grid features (DGF) are calculated by using the method of the UGF extraction but this time the search is starting from the bottom of the \( V(i) \) horizontal projection histogram. The output is again a ten element vector with binary values. Fig. 6(e) and (f) gives an example of the DGF extraction. This time the height of the extracted image, in our experimental set, was 50 pixels. Fig. 6(g) shows the final feature vector.

### 4. Comparison

The matching procedure can identify the word images of the documents that are more similar to the query word through the extracted feature vectors. Fig. 7 illustrates the comparison technique.

First, a descriptor is created by the seven extracted features as it is shown in Fig. 8. The first element is the weight to height feature, the second the image area density feature and the third the center of gravity feature. The following twenty (20) elements are the ones extracted from the vertical projection feature and the next fifty (50) from the top–bottom shape projection features. Finally, the last twenty (20) elements are the ones extracted from the upper and down grid features divided by 10 in order to...
prevent to overpower the others features. The rest of the features values are normalized from 0 to 1.

Next, the Minkowski $L_1$ distance is obtained between the descriptor of the query image word and the descriptor of each word in the database:

$$MD(i) = \sum_{k=1}^{93} |Q(k) - W(k,i)|$$  \hspace{1cm} (7)

where $MD(i)$ is the Minkowski distance of the $i$ word, $Q(k)$ is the query descriptor and $W(k,i)$ is the descriptor of the $i$ word.

Then the similarity rate of the remaining words is computed. The rate is a normalized value between 0 and 100, which depicts how similar are the words of the database with the query word. The similarity rate for each word is defined as

$$R_i = 100 \left( 1 - \frac{MD(i)}{\max(MD)} \right)$$  \hspace{1cm} (8)

where $R_i$ is the rate value of the word $i$, $MD(i)$ is the Minkowski distance of the $i$ word and $\max(MD)$ the maximum Minkowski distance found in the document database.

Finally, the system presents the documents that contain the words in descending order with respect to the corresponding rate. In our implementation, the documents presented to the user are those that have a similarity rate above 70.

The query word image is an artificial image that depicts the user query word and it is created by the proposed system with font height equal to the average height of all the word-boxes obtained through the word segmentation stage of the offline operation. In the implemented DIRS for our experimental set the average height is 50. The font type of the query image is Arial. However, the smoothing and normalizing of the various features described before suppress small differences between various types of fonts.

Finally, the created query image is being processed in exactly the same way as the document word images. That includes the application of preprocessing and feature extraction procedures.

5. Implementation

The proposed system is implemented with the help of the Visual Studio 2008 and is based on the Microsoft.NET Framework 3.5. The programming language used is C#.

The image documents that are included in the database are created artificially from various texts and then noise was added in order to implement in parallel a text search engine, which makes easier to verify and evaluate the search results of the DIRS system. Furthermore, the database being used by the implemented DIRS is the Microsoft SQL Server 2005.

The web address of the implemented system is the http://orpheus.ee.duth.gr/irs2_5. Fig. 9 shows a screenshot of the above program.

6. Evaluation

The precision and the recall metrics have been used to evaluate the performance of the proposed system. Recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database. Precision is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved. In our evaluation, the precision and recall values are expressed in percentage.

The evaluation of the proposed system was based on 100 document images. In order to calculate the precision and recall values 30 searches were made using random words. Table 1 shows those random words. The precision and recall values obtained are depicted in Figs. 10 and 11, respectively. The mean

<table>
<thead>
<tr>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. details</td>
</tr>
<tr>
<td>6. smoothing</td>
</tr>
<tr>
<td>16. period</td>
</tr>
<tr>
<td>26. diplomatic</td>
</tr>
</tbody>
</table>

Fig. 9. A screenshot of the implement DIRS.
precision and the mean recall values for this experiment are 87.8% and 99.26%, respectively.

The overall time for the above-mentioned 30 searches in our AMD Athlon 64 4200+ testing server with 2 GB of RAM is 11.53 s while the mean time for each search is approximately 0.38 s.

Furthermore, the same 100 document images were scanned from the FineReader® 9.0 (ABBYY FineReader®, 2007) OCR program, which translated all the characters of the document images into machine-editable text documents. Then, these text documents searched for the same 30 words of Table 1. The precision and recall values obtained are depicted in Fig. 12. The mean precision and the mean recall values are 76.667% and 58.421%, respectively, lower enough than the proposed system.

Then an experiment was made to test how the features of the proposed system react to the scaling of the world. The font height of the query word changed to double (100 pixels) instead of equal to the average height (50 pixels) of all the word-boxes obtained through the word segmentation stage of the offline operation. The precision and recall values obtained are depicted in Fig. 13. The mean precision is 89.315% and the mean recall is 97.593%.

Furthermore, in order to test the robustness of the features relative to the type of fonts, the query font was changed to “Tahoma” and the same 30 searches were made (Table 1). These two fonts have many differences without compromising the shape of the word. The precision and recall values obtained are depicted in Fig. 14. The mean precision and the mean recall values are 89.44% and 88.05%, respectively.

7. Conclusion

A document retrieval system was presented here. The proposed system makes use of document image processing techniques, in order to extract powerful features for the description of the word images. Seven meaningful features were used,
namely the weight to height ratio, the image area density, the center of gravity feature, 20 DCT coefficients extracted from the vertical projection, 50 DCT coefficients extracted from the top-bottom shape projection and 20 elements extracted from the upper and down grid. These features were selected in such a way that they describe satisfactorily the shape of the query words while at the same time they suppress small differences due to noise, size and type of fonts.

Our experiments were performed on a collection of noisy documents and gave mean precision 87.8% and mean recall 99.26%. The same experiment performed in the same database for a commercial OCR package gave lower results while experiments for different sizes and styles of fonts did not produce significant change to the performance.

Acknowledgments

This paper is part of the 03EΔ679 research project, implemented within the framework of the Reinforcement Programme of Human Research Manpower (PENED) and co-financed by National and Community Funds (25%) from the Greek Ministry of Development – General Secretariat of Research and Technology and (75%) from the E.U. – European Social Fund.

References