A Two-Stage Approach for Word Spotting in Graphical Documents

Arundhati Tarafdar, Umapada Pal, Partha Pratim Roy, Nicolas Ragot and Jean-Yves Ramel
CVPR Unit, Indian Statistical Institute, India
Synchronedia Lab, École de technologie supérieure, Canada
Laboratoire d’Informatique, Université François Rabelais Tours, France

Abstract—In graphical documents, presence of multi-oriented characters, connected characters with graphical lines or other character, intersection of text and symbols with graphical lines/curves is very common. As a result word spotting in graphical documents is still a challenging task that we try to solve (partially) in this paper. The proposed approach proceeds in two stages. In the first stage, recognition of isolated components is done using rotation invariant features and a SVM classifier. The characters having good recognition score and match in the query string are first selected for initial spotting. Because of structural complexity of graphical documents, we may miss some of the query characters during spotting in some documents. In that case, based on the position, size and orientation of the recognized characters in the input document image, regions where missing characters may be located (candidate regions) are defined. In the second stage, Scale Invariant Feature Transform (SIFT) is used to find those missing characters in the candidate regions for possible spotting. Finally using the positional, size, orientation as well as inter character gap information of the recognized components spotting is validated. Experimental results show that the method is efficient to locate a query word in multi-oriented and/or touching graphical documents.

Keywords—Document Image Analysis; Word Spotting; Graphical documents; Information Retrieval; SIFT

I. INTRODUCTION

With the rapid progress of research in document image analysis many applications are coming up to manage the paper documents in electronic form to facilitate indexing, viewing and extracting the intended portions. Recently, there has been much interest in the research area of Document Image Retrieval (DIR). DIR aims at finding relevant document images based on image features of user’s query. Word spotting, one of the important research areas in DIR looks for different instances of the query in the document images. Nowadays, when we want to retrieve documents, for example through internet, we do not want only to search inside books or images of books. We also want to look for graphical images like engineering drawing, maps etc. In such graphics documents the text or symbols appear along with graphical objects such as rivers, border lines etc. In maps, text lines appear frequently in different orientations such as curvy-linear other than the usual horizontal and vertical directions. The inter-character spacing in graphical documents differs according to annotation style. Because of such behaviors of graphical document, word spotting in graphical document is a difficult task.

Although there exist many pieces of work on text/graphics separation [4, 6, 7, 10, 12] and symbol/logo spotting [19], to the best of our knowledge there is no significant work on word spotting in graphical document. Rusiñol and Lladós [19] used local descriptor and proximity graph for word & symbol spotting, but that won’t work for different kind of multi-oriented documents. Roy et al. [2] used rotation invariant angular feature to extract multi-oriented characters and used character pair joining by dynamic programming for word spotting, but it can’t handle touching or broken characters. Some techniques [1,3] are also discussed about isolated character localization in graphical documents but they don’t deal with word spotting.

In this paper a two-stage approach has been proposed towards word spotting in graphical documents. In the first stage, using rotation invariant features isolated components are recognized and matched with the characters of the query string. Based on these matching results initial spotting is done. Because of structural complexity of graphical documents as well as touching and overlapping, we may miss some of the characters during initial spotting in some documents. In that case, based on the position, size and orientation of the recognized characters in the input document image, some candidate regions where missing characters may be located are defined. In the second stage, SIFT is applied to find those missing characters in the estimated candidate regions for possible spotting. Finally using the positional, size, orientation as well as inter character gap information of the recognized components spotting is validated.

The rest of the paper is organized as follows. In Section 2, we explain briefly the system’s overview. The detailed spotting scheme has been described in Section 3. The experimental results are demonstrated in Section 4. Finally, conclusion and future scope is given in Section 5.

II. SYSTEM OVERVIEW

A single page of graphical documents normally contain text characters printed in different fonts, sizes and orientation etc. to describe different entities. Thus, when these characters touch with graphical lines, it is not easy to recognize them. To take care of it, our proposed system works in a combination of straight forward approach for isolated characters and then
approaches to separate and locate connected text characters. We have used Otsu technique for document images binarization. At first the text and graphics portions are separated and isolated characters are extracted from text portions. The isolated characters are labeled using support vector machine classifier where rotation invariant features are incorporated. Based on the recognition results of the characters present in a graphical document image, the proposed system initially locates the position of different query characters (characters present in the query) in the image. If the system is able to locate all of the query characters in the image, maintaining some specified condition (discussed later) to be a word, then query word would be spotted there. Following the same procedure, if anywhere partial matching is found with query word, then it is assumed that there may be some missing characters (due to connectedness or broken part in the graphical document) in the document. So, depending on partial spotted positions of query word, probable candidate region of missing characters are detected. Next SIFT is applied in those region to find remaining missing characters of similar shape in the document for spotting. In Fig. 1 a block diagram of the proposed scheme is given.

Our detailed spotting approach has been divided into five parts. First isolated character recognition is done. Next initial query character spotting is done. If any query word spotted partially, the system finds some candidate region of possible missing query characters in next step. Then SIFT is applied in those region to find those missing characters. Lastly, validation of spotting is done. Details of these different parts are discussed bellow.

A. Character Recognition

At first text and graphical portions are separated from document images into two different layers using QGAR library [11]. Characters which are connected with graphical line will come under graphical layer and isolated characters will come under text layer. One such graphical document and corresponding graphical layer (after separated from text layer) have been shown in Fig. 2. Next connected component analysis is done in text layer to get individual components. Recognition of text characters in multi-oriented and multi-scale environment is a challenging task and different shape descriptors like Angular Radial Transform (ART) [14], Hu’s moments [15], Zernike moments [16], Fourier–Mellin [17] etc. are proposed to recognize such multi-oriented characters due to rotation, translation and scale invariance properties of these descriptors. Experimentally Fourier-Mellin feature outperformed among these different rotations invariant shape descriptors and hence Fourier-Mellin based features has been used in our work for isolated characters recognition using support vector machine (SVM) classifier. Details of training database used in SVM are discussed in results and discussion section.

Components having high confidence (greater than 0.5) in SVM prediction are considered as recognized characters. Components recognized with less confidence are considered as unrecognized. Fig. 3 shows an example of partially recognized text. We could see from Fig. 3 that SVM recognizes all characters except ‘N’ due to its touching with a line.

B. Initial spotting of query in recognized text layer

Now system has recognized isolated characters. Because of graphical documents, some characters may be connected with graphical or other text components as a results such connected characters may not be recognized. To reduce chances of false matching and also to make easy searching of the intended unrecognized texts, long lines (assuming that they are part of

Fig. 1. Flowchart/Block Diagram of our proposed approach

Fig. 2. (a) Sample map used in our experiment, (b) Graphics layer seperated from (a)

Fig. 3. Part of recognized text from Fig. 2 (red color text denoting recognized text)
graphic line not part of text) between any two junction points which are having bigger length compared to a character height or width have been removed from the graphical layer. For this removal purpose, first skeletonization of the graphical layer is done. Next the skeleton image is vectorized using QGAR [11] library code and long edges are removed from the image. In Fig. 4 two examples (denoted by red marks arrow) of such kind of lines shown which would be removed by this process. By this process almost all long graphical lines are getting removed, leaving only small lines and characters in that layer. Now image is ready to find remaining missing text characters and it is merged with the unrecognized portion (having low confidence in SVM) of text layer resulted from previous section. For the next processing, this merged document is named as document A and remaining text layer with recognized text is named as document B. We will refer such documents as A and B in the rest of the paper.

![Fig. 4. Graphical layer from Fig. 2 (b) (red arrows are shown examples of long lines between junction point which has been removed).](image)

Now query word is considered character-wise and sequentially for spotting. From document B (mentioned before), the system tries to find and spot query characters. Spotting is done recursively using k-nearest neighbor (k=6) method. Suppose we have query word ‘PRINSESSE’. According to Fig. 3 first character ‘P’ will be spotted, and then using k-nearest neighbor ‘R’ followed by ‘I’ will be spotted. Next, it will try to find ‘N’ but ‘N’ won’t be found as it belongs to document A (mentioned before) because of touching. So it will try to find ‘S’. Likewise next ‘E’ and ‘S’, ‘S’, ‘E’ would be searched and spotted one-by-one. While spotting, these characters are taken into consideration recursively if they have been followed three criteria: (a) distances between the CGs of two consecutive characters should be similar. (b) Two consecutive characters should be similar in size (c) angle between CGs of consecutive character pair with horizontal axis should be similar to the earlier pair.

Following above procedure a word can be spotted as candidate word if all query characters match. Otherwise if partial matching found with a query word, our system assumes that there may be some missing characters lie in between due to their touching or broken part. This spotting is called **initial spotting**.

### C. Candidate region detection of missing character

Here the context or spatial information of spotted texts has been used to find probable missing character’s region. In Fig. 5 multi-oriented word ‘PRINSESSE’ with touching (touching with graphical line) component have been shown. Here only the character ‘N’ is not recognized using method of isolated character recognition mentioned before. Remaining characters are spotted using k-nearest neighbor if query ‘PRINSESSE’ is used. Using recognized portion of a candidate word, the location of the query characters which are not spotted has been estimated and this located region is called **candidate region** of missing characters. After spotting of two characters, a rough angle (Aw) is estimated using the CG of recognized characters. Also spacing or distance (Dw) between them and an approximate height (Ht) and width (Wd) of each character are obtained from its bounding box (Bw) information. In our approach at least two query characters must be spotted during initial spotting. If missing characters lie in the middle of a word, then it is easy to assume their position. Using position of last spotted (Lp) character before missing characters and position of next spotted (Np) character after missing characters are taken. Starting from Lp a region up to Np having height (5*Ht), as shown in Fig. 5, is estimated from document A considering similar angular direction of Aw. This estimated region is the candidate region of the missing character. To avoid chances of not retrieval of any character, we consider maximum possible candidate region and hence we use the threshold 5*Ht. If any of the extreme characters of a word is missing then extreme portions with height (5*Ht), width (4*Wd) and similar angular direction (obtained from the two extreme spotted characters) are considered as candidate region.

![Fig. 5. For keyword ‘PRINSESSE’, where only ‘N’ is not recognized in first stage, (green dashed box- estimated candidate region)](image)

### D. Missing character detection using SIFT

Now SIFT is used to find query characters in the estimated candidate region of missing characters. If query characters are present in document B, then they are used as query images for SIFT, otherwise query characters taken from our character image database are used. SIFT is applied in the estimated region for finding missing characters of similar shapes. If all text characters or whole document image would be searched for particular characters, many false alarms would be found after applying SIFT [1] due to curvature nature of document. That is why SIFT is focusing on particular region. Scale Invariant Feature Transform (SIFT) feature usage for object recognition involves two steps - (1) SIFT feature extraction
and (2) Object recognition. Recognizing an object using SIFT can be performed using the Key-point matching. Each feature is composed by four parts: the locus (location in which the feature has been found), the scale, the orientation and the descriptor. The descriptor is a vector of 128 dimensions. In Fig. 6 we could see one query character spotting using SIFT and its keypoint descriptor regions.

E. Spotting word validation

Here we check the compatibility of the characters obtained by initial spotting made by recognition and the candidate region spotting made by SIFT. If the compatibility among the characters is satisfied then we mark that spotting is valid. Compatibility checking is done in terms of position, size and orientation of the characters present in the spotted area. After SIFT matching, if any missing character spotted in the estimated candidate region, then component’s rough bounding box (exact bounding box cannot be taken as it may have touching portion) is checked to see whether (i) its position is matching with other spotted characters (ii) its size matches with others and (iii) its orientation agrees with other. If all the above conditions are satisfied we declare the spotting is valid

IV. EXPERIMENT & RESULT

A. Dataset

We have considered 30 different real graphical maps and 20 different query words to test our method. As there are very few proper grayscale maps available, for our experimental purpose we have taken synthetic dataset used by Roy et al. [21]. There are approximately 35–40 words in each of the documents. Documents are digitized in grey tone at 300 dpi and we have used Otsu binarization technique for their two tone conversion. QGAR [11] library is used to extract characters from documents.

In our experiment both English uppercase and lowercase alpha-numeric characters are considered, so we should have 62 classes (26 for uppercase, 26 for lowercase and 10 for digits). But because of shape similarity due to orientation some of the characters like ‘d’ and ‘p’, ‘b’ and ‘q’, etc. are grouped together. Hence, in our approach we considered 40 classes of character shapes. Different fonts of characters including Times New Roman, Courier and Arial have been used to learn the SVM classifier for the experiment.

B. Experiment Details

A first part of the experiment is dedicated to the evaluation of the character recognition level. For that, we have considered four different sets of data consisting of multi-oriented and multi-scale text characters. One of the datasets is from graphical documents and its size is 2000 characters. The ground truth of this dataset has been generated manually for the performance evaluation. The other three datasets are synthetic data, constructed from Arial, Courier and Times New Roman font characters of font size ranging from 12 to 30. The size of all of these datasets is 5000. As said before, a comparison has been done with different rotation invariant feature descriptors used in the literature, namely: ART, HU, Zernike moments and Fourier–Mellin. As mentioned earlier, we got Fourier-Mellin as best rotation invariants feature which maintaining more than 95% recognition rate. So we have trained our SVM classifier with approx. 17000 isolated characters including real and ideal data featured with Fourier-Mellin. The SVM classification is done with 5-fold cross-validation. Some confusion is happening between ‘z’ and ‘zh’, ‘S’ and ‘5’, ‘D’ and ‘0’, ‘b’ and ‘h’, ‘t’ and ‘1’. Our system is not giving any false positive in case of query word of length more than 3, but we are missing few candidate words, especially when in graphical documents texts are not smooth and they are much noisy. In Fig. 7 we have shown few cases where our method is not able to spot query word properly.

To evaluate the performance of the system with query words against graphical document images, we have used precision (P) and recall (R) for evaluation of spotting of candidate words. Each spotted word images is considered as relevant or not, depending on the ground truth of the data. For a given spotted result, the precision measure P is defined as the ratio between the number of relevant retrieved items and the number of retrieved items. The recall R is defined as the ratio between numbers of relevant retrieved items to the total number of relevant items in the collection. Average precision and recall rate of our system is 83.8% and 93.4%.

![Fig. 7. Examples of a graphical document where our system could not spot for the query word “Udaipur”, “Mysore”, “BHUTAN” etc. for their poor samples.](image-url)
V. CONCLUSION & FUTURE WORK

In this paper a two-stage approach has been proposed for spotting in graphical documents. Initially a multi-oriented and multi-scaled text character recognition method has been used to generate high level local feature to take care of complex multi-oriented and multi-sized environment of graphical documents to identify isolated characters. We have adapted SIFT in this application to locate text characters which are touched/overlapped. To reduce overhead and false alarm we have chosen particular missing candidate region to apply SIFT. Our approach is giving acceptable performance, still we cannot rely on SIFT completely as there is a high chance in false alarm, also it has given some amount of not retrieval. In future, we plan to work on graph matching [20]. The recognition performance could be improved by removing the rotation ambiguities, by focusing more on candidate region. The challenge will then be to characterize each found ROI in a robust way to generate indexes (or codebook [18]) to describe the content of each image (image signatures). Statistical approaches like SVM or clustering or Key point techniques (SURF) can be used according to the type of the ROI.

ACKNOWLEDGMENT

This work is partly funded by IFCPAR Project (No 4700-IT-1).

References