A Robust Page Segmentation Method for Persian/Arabic Documents

M. Hassan Shirali-Shahreza[1], Sajad Shirali-Shahreza[2]
[1]Computer Engineering Department
Yazd University
Yazd, IRAN
[2]Computer Engineering Department
Sharif University of Technology
Tehran, IRAN
http://shahreza.shirali.ir http://sajad.shirali.ir

Abstract: Optical Character Recognition (OCR) softwares are widely used in the office automation systems. One of the first steps in the recognition of the documents is to segment the input image. Various methods have been offered for the English language. For the Persian/Arabic Language, however, no complete method has been found yet. In this paper we present a new page segmentation method for Persian/Arabic printed texts. This method has been inspired by the effect of the spreading of ink on paper. One of the most important characteristics of this method is its non-sensitivity to rotation.

Key-Words: - Page Segmentation, Image Analysis, OCR, Pattern Recognition, Persian/Arabic Document

1 Introduction

With the increasing application of the computer throughout the world, there has been an increasing need to digitize references and documents. One of the best ways to digitize the present documents is to use an OCR software. A first stage in the recognition of a document is to segment the input and identify the areas that contain text. The degree of proper segmenting and removing the non-text areas such as pictures are among the factors that contribute directly and considerably to the final accuracy and the quality of the OCR. Therefore, much work has been done for presenting a variety of methods for page segmentation [1]. Much research has been on English and proper methods with a good degree of accuracy have been achieved, and are presently used in commercial programs [2]. In the case of Persian/Arabic, however, because of the special characteristics of the Persian/Arabic script, no complete and appropriate method has been achieved yet, even for printed texts although relatively good methods have been provided in this respect [3, 4, 5, 6 and 7].

In this paper we introduce a robust method for segmenting Persian/Arabic documents that is to a certain extent different from the previous methods. The main idea of this method was inspired by the effect of spreading of ink on paper, considering the fact that, in printed texts, the distance between words and lines are proportional to the size of their font.

The organization of the paper is as follows: Section 2 expresses the special characteristics of the Persian/Arabic script, which make ineffective many of the methods that worked in the case of the English script. Section 3 discusses the ideas of our method and algorithm. Section 4 reports the advantages and disadvantages of our method. Section 5 reports the experimental results. Section 6 has some suggestion for further works. The conclusion is in the last section.

2 Characteristics of the Persian/Arabic Script

The Persian/Arabic script has unique characteristics that make it very hard to use the results obtained with other scripts. Some of these characteristics are [3, 4 and 7]:

2.1 The letters are connected to each other

The most important characteristic of the Persian/Arabic script that makes it much different from the English script is that the letters are connected to each other in the Persian writing system. In addition to the many problems in the recognition stage, there will also be limitations in the segmentation process as well. For example, in English writing, wherever in the text that the same font is used, the sizes of the connected component are very close to each other because the sizes of the different letters are almost the same. While, in Persian/Arabic writing, two connected components in the same line may have very different sizes. As you seen in Fig.1, the size of the second component of a Persian word is several times bigger than the first one. On the other hand, this creates a big difference in the dimensions of the components.

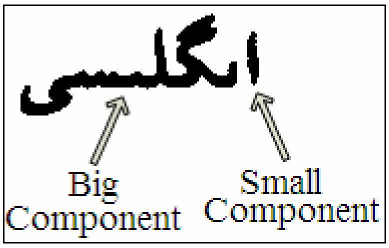


Fig. 1 Different connected component size in a Persian word

2.2 Dot and Sarkesh (Oblique Stroke)

The dot over and under letters and the sarkesh (oblique stroke over in the letter "گ") are other characteristics of the Persian script. The sarkesh does not exist in the English alphabet although the dot does. Even as to the dot, however, there is much different between English and Persian. In English, only the small letters "i" and "j" have a dot over them while in Persian 17 out of the 32 letters have dots, 2 of them having 2 dots, 4 having 3 dots, 11 having 1 dot each. In general, there are many dots in a typical piece of Persian written material. On the other hand, if the dot over "i" or "j" is removed, it will still be recognizable to a great extent while in Persian there are many letters that are different from each other only in the number or position of the dots. This is important because the size of the dots is very small and, if special care is not taken during the noise-removing operation, a large number of dots will be removed. On the other hand, because of the dots, some noises are not removed due to their similarity to the dot and this retains some noise and reduces the quality of segmenting. Figure 2 shows the similarity between a noise and a dot.

2.3 Right-to-Left Direction

Unlike English, which is written from the left to the right, Persian/Arabic is written from the right to the left. As a result, the right and left margins in Persian/Arabic texts is opposite to those in English texts, and the right margin is bigger in Persian/Arabic writing.

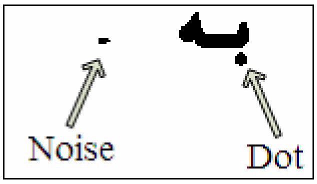


Fig. 2 Similarity of a Dot and a Noise

3 Suggested Method

3.1 Idea

The main idea of our method is inspired by the spreading of ink on paper. When ink is used for writing on paper, if the paper is of a material that easily absorbs liquids, the ink will be spread around the written words and the words will look a bit thicker. The degree of spreading of ink around the word depends on the size of the word, i.e. the amount of ink used. This inspired the idea that, if the degree of spreading is high, it will make the letters of a word and the words of a line connect to each other and a connected component will be created. In a more general outlook, if the amount of spreading is appropriately determined, it can connect the lines of a paragraph too.

3.2 The Algorithm used

As it was mentioned above, the purpose is to simulate the effect of spreading of ink on paper. We also want such amount to be in proportion to the size of that piece. Also we consider the fact that the distance between the lines, words and the different parts of a word depend on the font with which the text was written.

In this section, the details of our algorithm are described. The outputs of stages on image Fig.3 as input are shown in Fig.4 to Fig.7. In figures that are needed to show distinct components, different gray scale colors are used.

3.2.1 Preprocessing the image

As in most OCR methods, the assumption in this project was also that the input is a scanned black-and-white (B/W) image. Therefore, the image contains a lot of noise. Two major jobs are done in the preprocessing stage: removing the noise and removing the big components which are almost certain not be part of the text.

To do these two jobs and considering the fact that, in the next stage, there is a need to the connected components and the size of each, first the connected components of the image are specified by using a classic algorithm [8]. Then the number of the black pixels in each component is counted. The component whose number of pixels is less than a certain fixed amount (which is obtained by studying a number of sample documents), is identified as noise and removed. The selection of this fixed amount is very critical because, as it was mentioned before, the size of a dot is very close to the noise created during scanning.

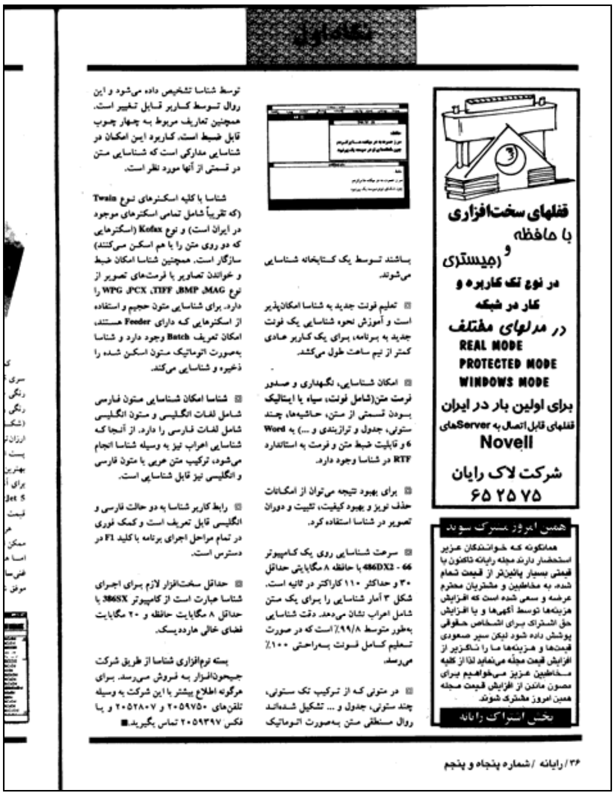


Fig. 3 Sample Persian document use as input

On the other hand, if the number of pixels is greater than another fixed amount (which is also obtained by studying a number of sample documents), the component is identified as non-text and removed. In this project, by selecting an appropriate value for this fixed number, a large number of big non-text components such as borders and the main parts of the figures are removed. Attempt was made in the project to obtain such threshold values dynamically for each image.

However, there was little improvement in the practical results and, considering the large overhead; the decision was finally for fixed thresholds.

In addition, to increase the speed, the component-labeling algorithms, which actually require two pass scanning of the image, was changed so that, simultaneously with the finding of the connected components and with a little additional work, the counting operations of the number of pixels in each component and the removing of the noise and the non-text parts are also carried out. The result of preprocessing stage on Fig.3 is shown if Fig.4.

3.2.2 Simulating the Ink Spread Effect

At first, for each component, considering the number of its black pixels, a spread radius was defined. According to the study carried out, the following formula (1) was obtained and used.

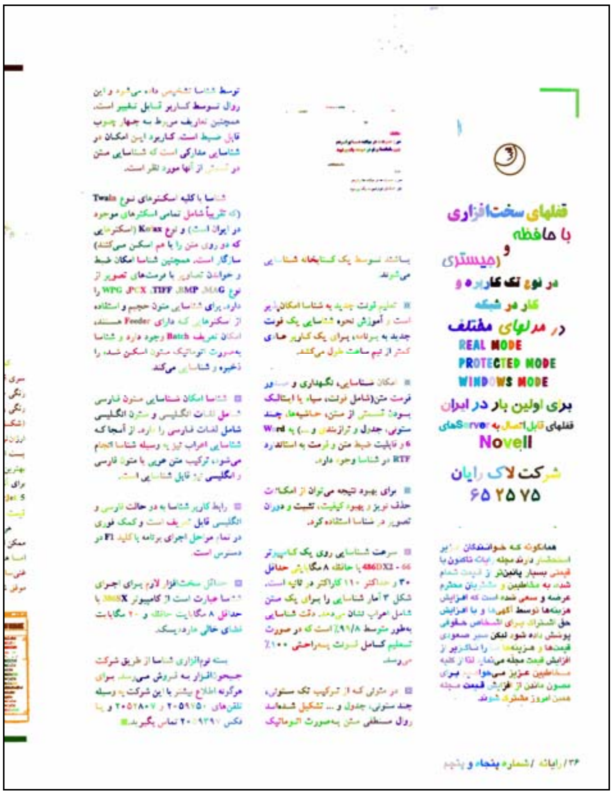


Fig. 4 The input image after preprocessing

$$\text{Radius} = (\text{Num. of black pixels in component})/\ (20) \qquad (1)$$

It was found out in the studies that the bigger fonts were used for titles in most cases and the titles were single-line mostly. Also, the distance between the title and the main text is almost equal to the distance between two lines when using the title font. Therefore, if the formula relating to the smaller fonts is used for bigger fonts as well, the title will connect to the main text and the two parts that should be separate will be merged and form one part, which is not desirable. Therefore, a separate formula (2) is used for bigger components.

$$\text{Radius} = ((\text{Num. of black pixels in the component})/\ (20))*(0.4) \qquad (2)$$

For each black-edge pixel (a pixel that has a white neighbor), a circle is drawn that has this pixel as its centre and the spreading radius in black. This somehow simulates the ink spread because the black pixels project outwards the edges to a certain extent. In this project, the square was tested instead of the circle. Although this improved the speed of the work, it reduced the accuracy; so we use circle. A further step in accelerating the program was to draw a quarter of circle for each black-edge pixel in the direction having a white neighbor because it was seen that many of the black-edge pixels had 1 or 2 white neighbors. Theoretically, this can improve the speed of this part for more than 50%.

After applying this stage on Fig.4, Fig.5 is produced.

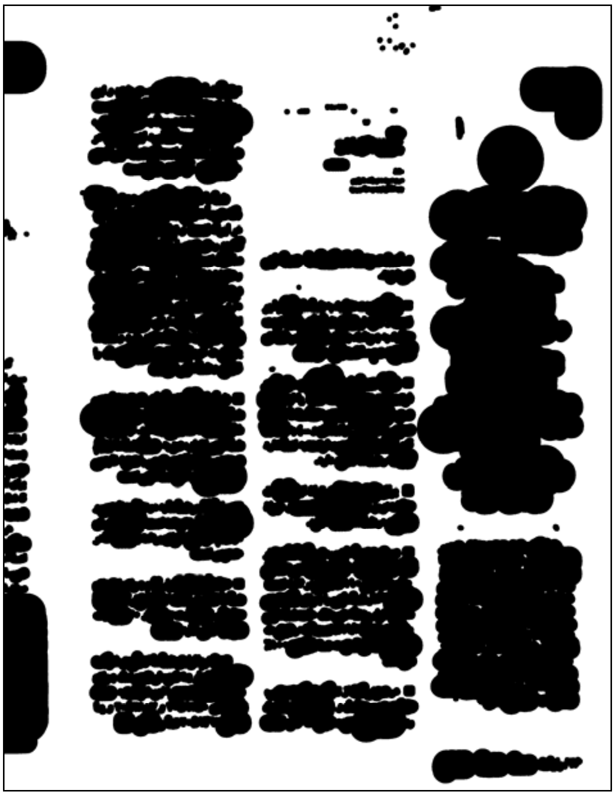


Fig. 5 Input image after simulating the ink spread effect

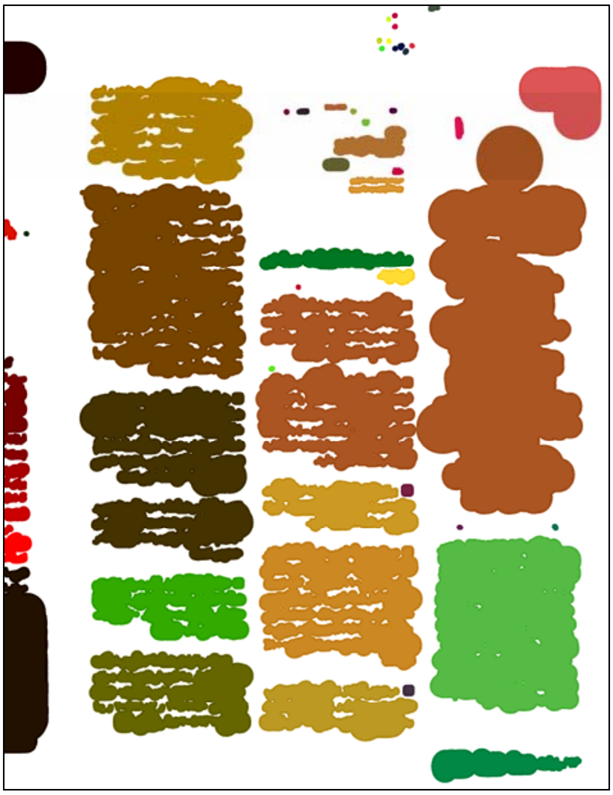


Fig. 6 Mask generated for making output

3.2.3 Identifying the Parts of the Image

After spreading, the connected components are once more identified. Each of the connected components obtained in this part identify a certain part of the main image. In this part, the algorithm which was expressed in section 3.2.1 is used for identifying the connected components.

The output of this stage on Fig.5 is shown in Fig.6.

At last, the connected components of the resulting image are used as a mask for the image obtained from stage 3.2.1. The black pixels of the image from stage 3.2.1, which is located in the same component in picture from stage 3.2.2, is labeled as a part of the image. This will be the output of the program.

Using Fig.6 as mask for Fig.4 yields Fig.7 which is the final output of the algorithm.

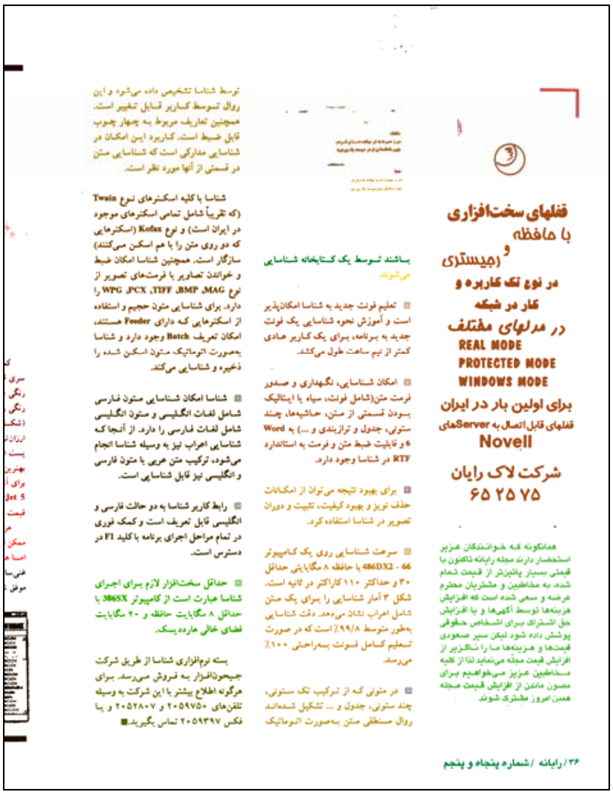


Fig. 7 Final Segmentation Result

4 Advantages and Disadvantages

This method has some advantages and disadvantages of its own. In this section, the advantages and disadvantages of the method are examined.

4.1 Advantages

4.1.1 Non-Sensitivity to Rotation

As most inputs in an OCR software are scanned documents, the images usually rotate a bit. Other than the usual sensitivity of the identifying operation to rotation, many segmentation methods such as the rectangle-making method are also sensitive to rotation. Indeed the degree of such sensitivity is different from method to method but usually most methods do not work properly with more than 10 to 15% rotation. Although malfunctioning in rotation of more than 15% seems reasonable [9], if a method is not sensitive to rotation, this will be a strong point for it. Indeed it should be considered that in the methods that bear up to 15% rotation, their accuracy usually decreases by increasing the degree of rotation.

In this method, because of the idea that was used and the algorithm that was presented and implemented, the program has no dependence on rotation and, as it was tested, even if the input image is rotated or made symmetrical to one of the axes, it will not affect the accuracy of the program.

4.1.2 Identifying Text with Border

In some methods, like [6], parts of the text that are within borders are removed as non-text. In our method, considering the thresholds used for identifying the non-text parts, in most cases the text borders were removed and the text itself was segmented properly.

4.1.3 Identifying Texts in Gray-Background Areas

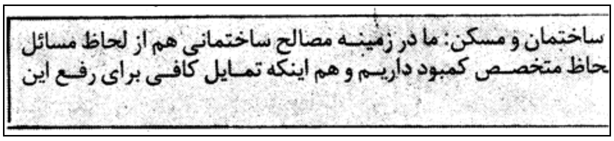
As the input in this method consists of binary B/W images, when the areas of the document where the text appears on a non-white background, such as gray, while converting to binary B/W image, the background is changed into close-distance black dots. Only some methods are able to identify this type [9]. In tests out by using our method, it was seen that these type of texts were usually recognized and segmented correctly, which is due to effective removal of the noises in the preprocessing stage.

For example applying this method to Fig.8 produces the output that is shown in Fig.9.

4.1.4 Recognizing Texts within Pictures

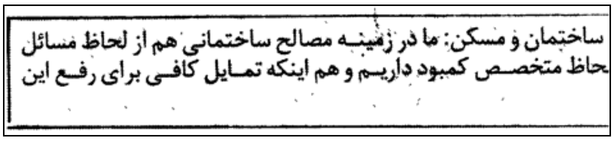
The input image may contain texts within pictures. In this method, because of method that used for removing non-text components, usually all the pictures are not removed although this is also considered as a weak point of this method. However, the texts within pictures will be recognized and segmented appropriately.

For example, numbers and text that are inside Fig.10 are segmented correctly as shown if Fig.11.



ساختمان و مسکن: ما در زمینـه مصالح ساختمانی هم از لحاظ مسائل
حاظ متخصـص کمبود داریـم و هم اینکه تمـایل کافـی برای رفـع این

Fig. 8 Text in gray background



ساختمان و مسکن: ما در زمینـه مصالح ساختمانی هم از لحاظ مسائل
حاظ متخصـص کمبود داریـم و هم اینکه تمـایل کافـی برای رفـع این

Fig. 9 Output of segmenting Fig.8

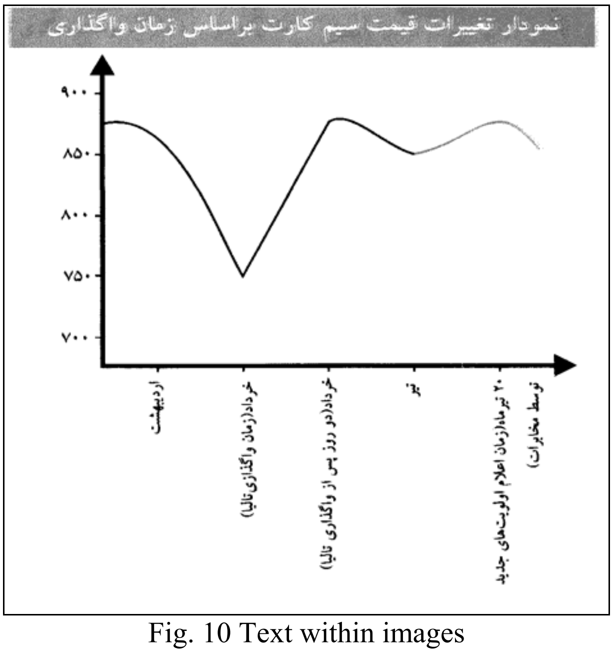


Fig. 10 Text within images

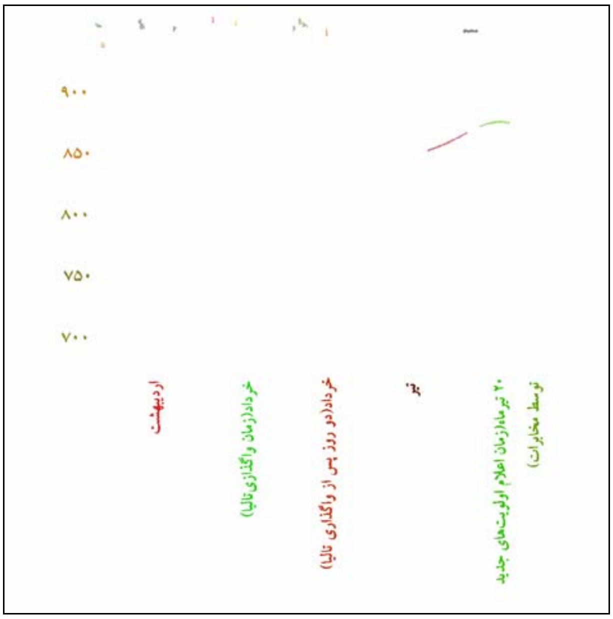

۹۰۰
۸۵۰
۸۰۰
۷۵۰
۷۰۰
اردیبهشت
خرداد(زمان واگذاری تالیا)
خرداد(دو روز پس از واگذاری تالیا)
۲۰ تیرماه(زمان اعلام اولویت‌های جدید
توسط مخابرات)

Fig. 11 Output of segmentation of Fig.10

4.1.5 Acceptable Results for Other Languages

Although this method was specially designed for Persian/Arabic and many characteristics of the Persian/Arabic script were considered in it, the tests on some samples from English, which were segmented with this method, also rendered acceptable results. This indeed requires further studying and testing.

4.2 Disadvantages

4.2.1 Slow Speed

This method is categorized as a bottom-up method. For each picture, each pixel is read, examined and written several times, so it has not a fast speed. This is one of the major disadvantages of this method. However, considering that the OCR software is usually used for non-real-time applications, it seems that this method can be improved into a commercially usable method.

4.2.2 Requiring a Big Memory

At least two copies of the image need to be stored in the memory to be used during the operations. Although this is not a big weak point because of the large volumes of memory available in computer systems, this may limits the application of this method for embedded systems.

4.2.3 Not Removing Some of Non-Text Parts

Some of non-text parts are not removed in this method. This can create problems for the OCR system. In this method, usually some small parts of an image may be remained in the output as a text region. In this project, many attempts were made to remove these small components. However, considering that in some tests some of this small components were actually part of the text, no proper method for removing this problem was obtained. Indeed the failure to remove all non-text parts has an advantage, which is specifying and segmenting texts within pictures, which was mentioned in Section 4.1.

5 Experimental Results

The algorithm was tested on a database consisting of 40 images. These images were scanned at 300 DPI and in gray scale mode. Then the scanned images converted into binary format. We used a variety of Iranian printed materials such as newspapers, sport journals, scientific journals, books and advertisement brochures for testing our algorithm.

Our program is written in Visual C++ using Microsoft Visual Studio .NET in the Windows XP Operating system. It run on an AMD Athlon 64 3200+ PC system with 1 GB RAM.

A majority of documents were segmented in less than 20 seconds. Small numbers of documents were segmented in about 25 seconds. In our test documents all of the texts in the boxes or with gray background were segmented correctly. As mentioned in the disadvantages, small part of some images were not deleted and appeared as small components in the output.

There were only a few components that merged incorrectly but some isolated words of some paragraphs were segmented separately.

We tested the algorithm on some rotated documents, and the rotation has no effect on the accuracy of the segmentation.

A lot of texts within images were recognized and segmented as text components.

6 Suggestions

One of the disadvantages of this method, which was already mentioned, is the failure to remove all non-text parts, which leaves room for much more work. To complete this method, before each component presented in the output should be examined and, after ensuring it is text, be put in the output.

Another problem is the slow speed of this method, which can be improved. The program that is now implemented has not been optimized so as to be easily changeable and new ideas may be easily tested in it. Some sophisticated algorithms, however, were used for increasing the speed.

Another part that requires further work is the thresholds that were used. Presently, the thresholds have been obtained manually and experimentally by using a number of samples. However, the thresholds may be optimized for a series of documents with similar characteristics, such as the pages of a book.

7 Conclusion

In this paper we present a robust method for segmenting Persian/Arabic documents. This segmentation method is a bottom-up method and is not very fast. Yet, because of its capabilities, such as non-sensitivity to rotation, recognizing texts within borders and pictures and texts within gray background, it can be used as an effective method. The tests that were carried out yielded satisfactory results but they were not ideal. This method and idea require further and more precise work to be improved to be a perfect method for commercial use.

References:

[1] S. Mao, A. Rosenfeld and T. Kanungo, "Document Structure Analysis Algorithm: A literature Survey", *IBM Almaden Research Centre*, 2002, http://0-lhncbc.nlm.nih.gov.csulib.ctstateu.edu/lhc/docs/published/2003/pub2003015.pdf.

[2] S. Mao and T. Kanungo, "A Methodology for Empirical Performance Evaluation of Page Segmentation Algorithms", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol.23, No.3, 2001, pp. 242-256.

[3] M. H. Shirali-Shahreza, *"Off-line Recognition of Farsi Handwritten Words & Numerals by Neural Networks"*, Ph.D. Dissertation, Electrical Engineering Department, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran, 1996.

[4] M. J. Tabrizi and M. H. Shirali-Shahreza, "A New Page Segmentation Method for Persian Documents", *Proceeding of the 15th IASTED International Conference on Applied Informatics*, Innsbruck, Austria, 1997.

[5] M. Radmehr and E. Kabir, "A page Segmentation Method for Farsi Documents", *Proceeding of 5th CSI Computer Conference (CSICC'2000)*, Tehran, Iran, 2000, pp. 52-59.

[6] K. Hadjar, O. Hitz and R. Ingold, "Newspaper Page Decomposition using a Split and Merge Approach", *Proceeding of 6th International Conference on Document Analysis and Recognition*, Seattle, USA, September 2001, pp. 1186-1189.

[7] K. Hadjar and R. Ingold, "Arabic Newspaper Page Segmentation", *7th International Conference on Document Analysis and Recognition (ICDAR 2003)*, 2003.

[8] R. Haralick and L. Shapiro, *Computer and Robot Vision*, Vol. I, Addison-Wesley, 1992

[9] A. Antonacopoulos, "Page Segmentation Using the Description of the Background", *Computer Vision and Image Understanding*, Vol.70, No.3, June 1998, pp. 350-369.