Using critical points in contours for segmentation of touching characters

Ventzislav Alexandrov

Abstract: *In this paper is presented a new method for segmentation of touching characters using critical points in their contours. The touching regions are determined based on distribution of the critical points along the contours. Critical points which are candidates for touching places are selected based on geometrical and structural information. Segmentation is completed by cutting the touching character objects at places defined by candidate critical points. Verification and correction of the selected candidates for cutting is performed by the recognition of the separated objects. Actually, this method represents a coupled segmentation-recognition process. The proposed here technique is tested on machine-printed characters but many of its ideas may be used for segmentation of handwritten characters too.*

Key words: *Character recognition, OCR, Character segmentation, Touching characters, Broken characters, Contours, Critical points*

INTRODUCTION

The majority of errors in the results of OCR systems is due to the image degradation of the scanned characters. Contemporary OCRs cope very well with different fonts and font attributes but if the character structure changes or the received character images become either touching or broken, the recognition rate falls considerably and it turns out that such characters are the major challenge for the present-day OCR systems.

The main causes for receiving images with connected characters, which cannot be recognized trivially, can be summarized in the following:

- Bad quality of the print
- Use of proportional fonts with serifs
- Not appropriate scanner settings

This paper presents a new approach for segmentation of touching characters which is based on contour analysis and especially on the defining of the critical points along the character contours which is presented in detail in [4]. It is shown here that the concave critical points are the places where touching character patterns can be separated. The convex critical points are the places where broken character patterns can be connected.

Algorithms and methods presented in the paper are tested on machine-printed text but as you will see below, some of the ideas may be used for handwritten text too.

SEGMENTATION METHODS IN LITERATURE

Several different techniques for segmentation of touching characters were reported in the literature. The most common is the technique based on vertical projection of the text rows. However, vertical projection has to be helped by other techniques and additional instruments for separation, such as the top and bottom profiles of the character patterns [2]. Another method is based on so called sliding window [1]. Very often this method gives erroneous recognition if part of connected characters looks like other normal characters. Sliding window is moved by step of 1 pixel at a time and the computational load can become very high. This may slow down drastically the recognition process. In [3] was proposed a new way for finding places where characters touch to each other. It is based on an AND function which is calculated along every two neighboring pixel columns and the results for every pixel column are summed vertically.

In the same paper is claimed that this AND function approach gives better result for finding the touching places than the vertical projection (fig1 a and b). Indeed, the vertical

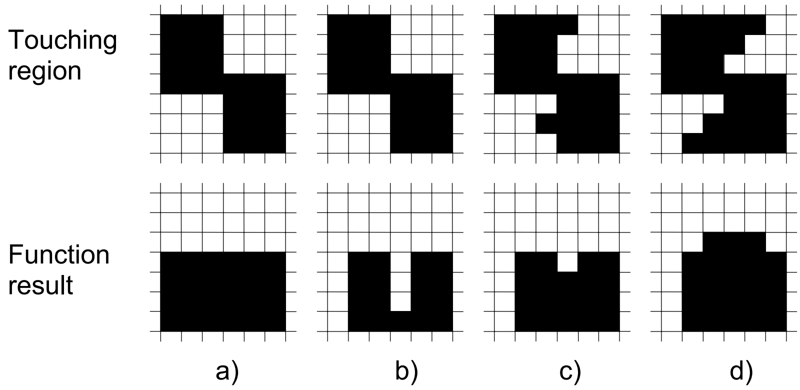

Fig. 1 Results of vertical projection and AND function

projection in fig. 1a doesn't give any clue for touching, while the result of the AND function summing for the same touching pattern gives significant notification that there are touching objects, fig. 1b. However, we discovered that even for this simple example pattern, the proposed AND function based approach fails if noise is present (fig. 1c) or pattern is oblique (fig. 1d). It became clear that this technique will not give satisfactory result in practice when noise is present or when the font is italic.

BASIC PARTS OF SEGMENTATION-RECOGNITION PROCESS

Before doing recognition of a given text, the symbols or the text objects have to be segmented and this procedure partitions the images of text blocks, lines and words into individual characters which are then submitted to the recognition procedure. In fact, most (if not all) of the character recognition methods and algorithms rely on successful segmentation. If segmentation fails, it is very unlikely that a correct recognition result will be received. On the other hand, verification of the cutting or connecting locations can be done only by recognition. If the recognition after the first cut fails, another cutting location is searched and the procedure of cutting and recognition is repeated. This process continues until the recognition of the whole touching character image is successful or no more cutting locations are left.

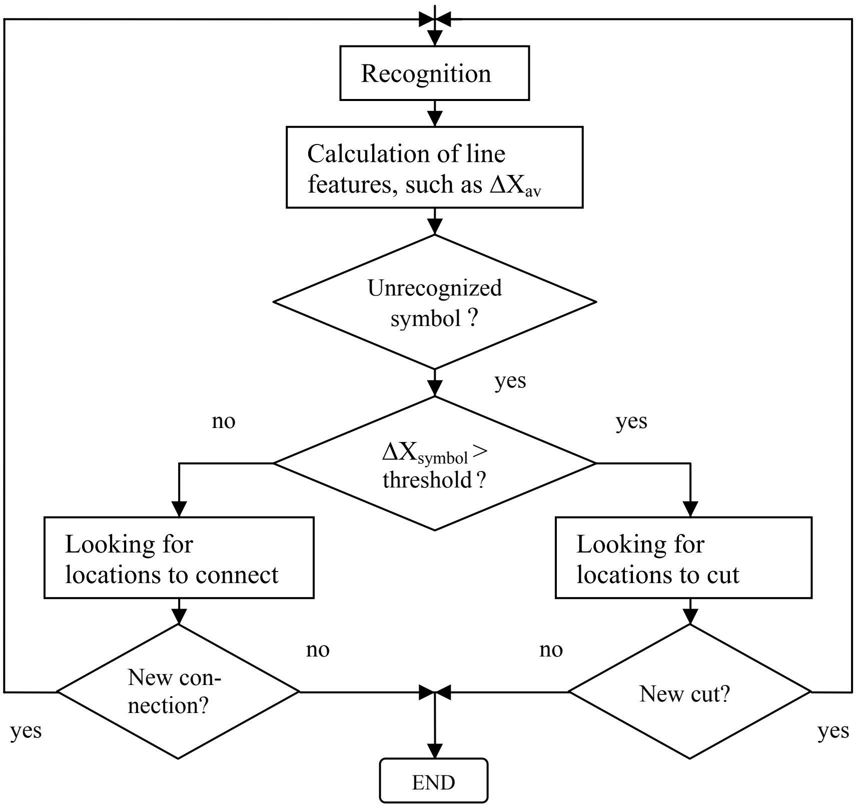

Fig. 2 Segmentation-recognition process

On practice, as it can be seen on fig. 2, cutting of the touching characters has to be combined with a module for artificial connection of the broken characters or the images received from the cutting procedure. Sometimes, it happens to cutting location to be not appropriate. In this case the connecting module should try to connect the wrong cutting. We discovered that the connecting module can't be ignored by simply restoring the original image because it will be not clear at which iteration is the procedure and how many cuttings or connections have already been done neither if they are good or not. And vice versa, in some cases, cutting module could repair wrong connections. It has to be emphasized that we can't expect to have successful recognition after each successful cutting or artificial connection. Character image may need more cuttings and connections for being successfully recognized. The algorithm ends when either all character images are recognized or no more cuts or connections are made. For escaping of infinite loops, criteria for forbidding some cuts are developed, such as number of cuts at the same place.

SELECTING APPROPRIATE CRITICAL POINTS

In contour analysis, contour boundaries of the characters are usually encoded either with 8-connected or 4-connected chain codes (also known as Freeman's codes). For the work described in this paper, 8-connected chain coding was used (fig. 3). In [4] can be seen how the critical points are selected and how their number is reduced and also how the contour noise is filtered for not bothering the recognition. Here only briefly will be pointed out that the critical points can be concave or convex and that the cutting place candidates are determined by the concave critical points. It also has to be emphasized that not all concave critical points represent cutting place candidates. Such candidates are only the concave points contained in up-turned (∩) or down-turned (∪) pocket like structures. Concave points contained in left-turned (⊂) or right-turned (⊃) pocket like structures are not cutting place candidates. The reason for this is the fact that we write in horizontal lines and we expect that a given character can be connected only to its right or left neighbour, not to the neighbour above or below. If our writing were vertical we wouldn't have the right to ignore the left-turned and right-turned pocket like structures as cutting place candidates. On fig. 7 can be seen in reality these pocket structures.

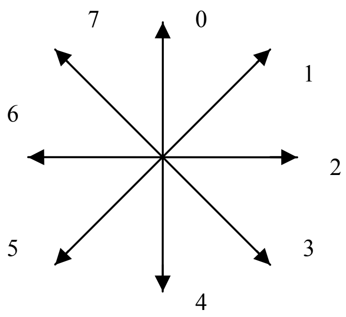

Fig.3 Freeman's codes

If contour is followed in clock wise direction, the concave point chain structures shown on fig. 4 represent the main part of all possible down-turned (a) and the up-turned (b) pockets which are of interest for character separation. The two ends of the shown structures can be both concave and convex. When an end is concave and matches the rules shown in flowchart on fig.5, it is also part of the pocket structure and can be regarded as cutting place candidate.

a) b)

Fig. 4 Chain code representation of all up-turned and down-turned structures

If numbering of chain vectors is selected as it is shown on fig. 3, then the main idea in the algorithm for determining the critical points belonging to up-turned and down-turned pockets can be followed in the flowchart on fig. 5.

There have to be at least two consecutive concave critical points (CP) in order to check their vectors (V) for appropriate directions. These directions determine surely if a given concave point is part of an up-turned or down-turned pocket. There is only one exception when no two consecutive points but only one single concave point is checked and this is the case when its difference chain code is bigger than 1. This exception can be seen in fig. 4, the last concave point chain structures for up-turned and down-turned pockets.

After observing and processing hundreds of touching characters, several simple rules which determine the order of cutting for touching character image or the order of cutting candidates points were drawn. As it can be seem, rarely only one cutting point candidate is present. When the candidates are more than one, they have to be ordered according to their probability to be the right place for cutting. The first cut is performed on the most probable place while the less probable place is left as last. In practice, three rules turned out to be enough and they are the following:

1. Advantage for the candidate with the best place along X axis
2. Advantage for the candidate with less pixels for cutting
3. Advantage for the candidate located closer to the top or the bottom of the image

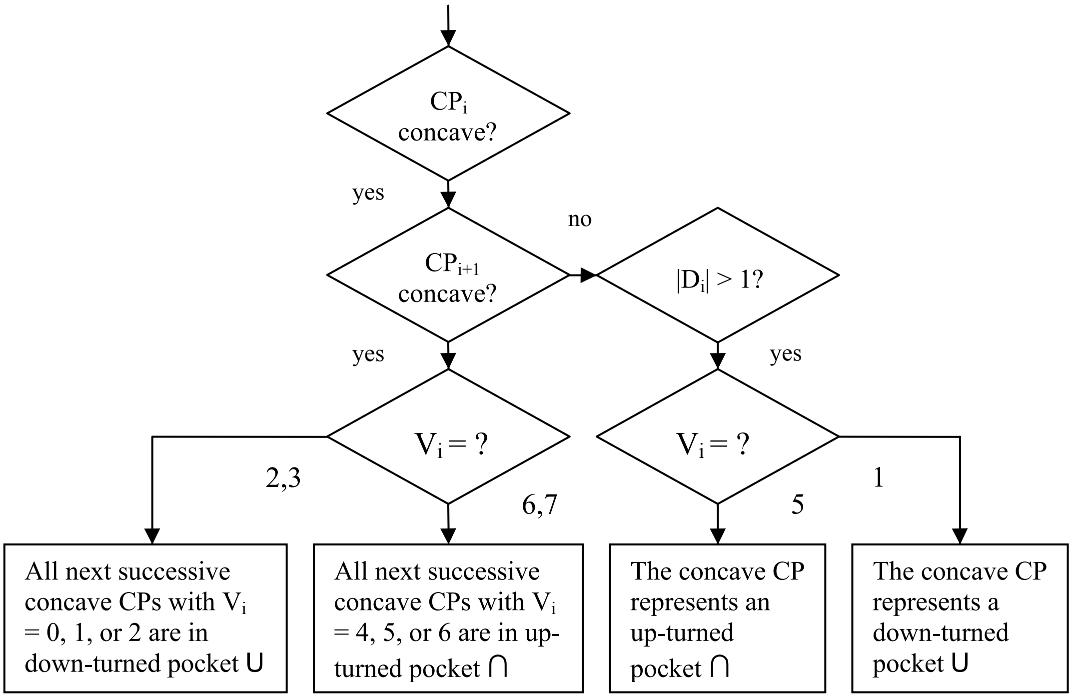

Fig. 5 Finding of up-turned and down-turned structure sequences

For determining the best place along the X axis, a special cutting cost function was developed (see fig 6). With the help of this function, to each cutting candidate point is assigned a value. This function is defined for every x point along the ΔX size of the touching characters. It is less probable for the real touching places to be located closely to the left or the right side of the connected image. Concave critical points which are located around the distance of ΔX_{av} (average ΔX size of the recognized symbols in the text row) from the left or the right side of the connected image have the biggest probability to be real touching points. That is why at these places the function has maximum values. From the practical point of view (especially for wide letters such as ж, ш, щ) it turned out that this function is better to have flat maximum value between the right and the left ΔX_{av} distance. The critical points for which this function has bigger value are the first places to cut.

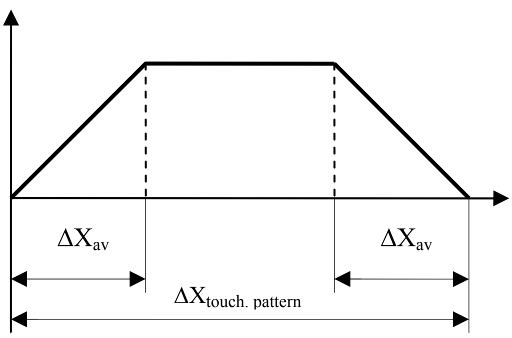

Fig 6 Cutting cost function

If for two or more cutting candidate points this function has the same value, then the candidate with smaller number of pixels for cutting has an advantage (the second rule). And if there are two or more candidates with the same number of pixels for cutting (their function values are the same too), then the candidate which is located closer to the top or the bottom of the image has an advantage (the third rule).

Sometimes, when the concave points of one up-turned or down-turned pocket are located at big distance from one another, an artificial concave point has to be introduced. Very often this is the place where the cutting of the object gives successful recognition.

CONCLUSIONS AND FUTURE WORK

The classifier used for this work is robust enough and it recognizes the characters even when not the best cutting candidate is chosen but some of its neighbours. In fact, it is more important to select the right pocket. On fig. 7 is shown an image of two touching characters and all its pockets and critical points. Generally, such kind of touching is not easy to separate because the mentioned in the beginning methods do not give good results, but with the help of the critical points it is fairly easy. One of the main advantages of the proposed method for separation of touching characters is the fact that the cuts are done not along the whole height of the connected characters, but locally, only at place of the concave critical points. If the cuts are produced from top to the bottom (along the whole height), then there is a big probability for involuntary breaking of the characters into unwanted parts. This certainly will be a trouble for the classifier.

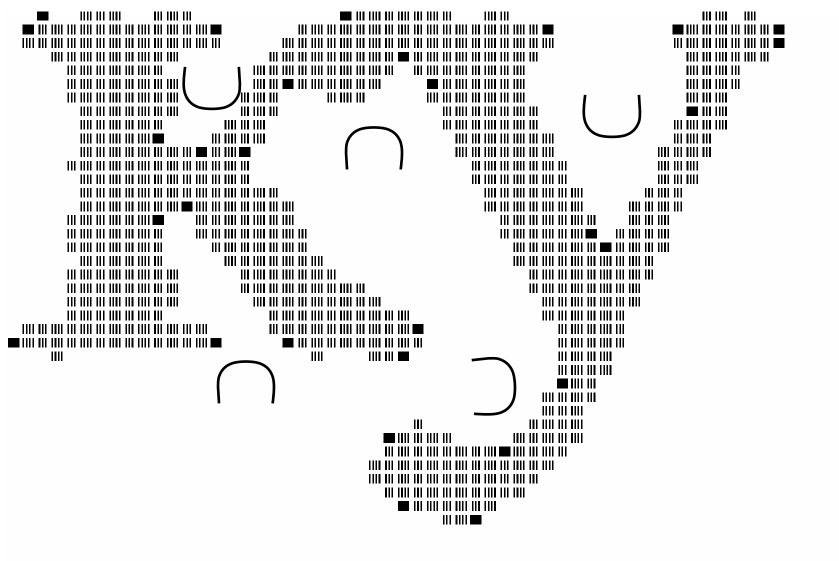

Fig. 7 Pocket like structures

In this paper has been presented a new technique for segmentation of touching characters which uses structural and geometrical features and the segmentation itself is verified and corrected according to the partial recognition result. It was pointed out that the touching regions have in most of the cases a well defined nature and that not all critical points can be regarded as cutting place candidates. The question about the hole contours and their cutting place candidates was not covered here but holes play an important role for separating touching characters.

Our next efforts will be mainly directed toward improving and refining the performance of the connecting module which has to solve the segmentation problems encountered with the broken characters.

REFERENCES

[1] Casey R.G., G. Nagy, Recursive segmentation and classification of composite character patterns, Proc. 6th Int. Conf. Pattern Recognition, Munich, Germany, pp 1023 -1026, 1982

[2] Liang S., M. Shridhar, M. Ahmadi, Segmentation of touching characters in printed document recognition, Pattern recognition, Vol. 27, No 6, pp.825-840, 1994

[3] Tsujimoto S., H. Assada, Resolving ambiguity in segmenting touching characters, 1st Int. Conf. On Document Analysis and Recognition, Saint-Malo, France, pp.701 – 709, October 1991

[4] V.Alexandrov, Using contours for recognition of alpha-numerical symbols - Proceedings of "Автоматика и информатика 99", Sofia, Bulgaria, Vol. 2, pp.113 -116, 19-20 June 1999

ABOUT THE AUTHOR

Ventzislav Alexandrov, MSc, Institute of Computer and Communication Systems, Bulgarian Academy of Sciences, Phone (+352 2) 732 951 (ext.119), e-mail: ventzi@bas.bg