Segmentation of Touching Characters in Formulas

Masayuki Okamoto, Syougo Sakaguchi, and Tadashi Suzuki

Dept. of Information Engineering, Shinshu University
500 Wakasato, Nagano, 380-8553, Japan

Abstract. Segmentation of touching characters in mathematical formulas needs a different method from the one for text lines, because these characters may occur in horizontal, vertical or diagonal directions.Our segmentation method is based on the projection profiles of a given binary image and minimal points of the blurred image obtained by applying the Gaussian kernel to the original image.

1 Introduction

Formula recognition is of interest in digitization of scientific documents. We have proposed two types of methods for formula recognition [1,2]. This paper describes the extension of our methods for formulas which contain touching characters. A number of segmentation methods of touching characters in text lines have been proposed in the literatures. These methods can be applied for character strings printed in only one direction; horizontal or vertical (e.g., in Japanese documents). In mathematical formulas, characters or symbols are placed in horizontal, vertical or diagonal directions. This fact means we need a new segmentation method for the touching character problem in formulas.

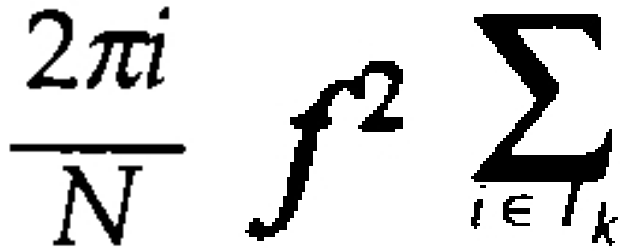
It is common to use features based on a vertical projection in the character segmentation of text lines. The horizontal and vertical projection profiles are also useful to estimate positions for character segmentation in formulas. However most characters or symbols in formulas are printed in slanted form, there are many cases where touching characters can be separated adequately only by a slant line not a horizontal or vertical line. This line can be estimated from minimal points of the blurred image obtained by applying the Gaussian kernel to the binary image. These points are located at the valley between blurred blob components which correspond to each character.

Since the decision whether a given blob is touching characters or not is very difficult, we use a simple criterion that if a classifier rejects the blob, we consider the blob consists of two characters. Then both of horizontal and vertical segmentation are tried, and the classification for each separated component is carried out again. Finally, the average scores for both of the components are compared between two segmentations, and the one which have the higher score is adopted. In this paper, the type of touching characters we examine and the segmentation algorithm are described in next Section 2. Experimental results and performance issues are discussed in Section 3.

S.-W. Lee and Y. Nakano (Eds.): DAS'98, LNCS 1655, pp. 151–156, 1999.

2 Segmentation

In our formula recognition system, before structure analysis of a formula, each blob of symbol or character components as a single connected component is classified. At this step, if the score of classifier is less than a priori value, then we assume the blob consists of touching characters. As shown in the Fig.1, typical occurrences of touching characters are grouped into three groups; horizontal, vertical or diagonal adjacency. Although the relative positions of horizontal and diagonal touching characters are different, they can be separated horizontally. Then, from now on, we will examine a segmentation method which separates touching characters in horizontal and vertical directions.



Horizontally adjacent Diagonally adjacent Vertically adjacent

Fig. 1. Typical occurrences of touching characters

Fig.2 shows a binary image of blob with its horizontal and vertical projection HP and VP defined as the functions mapping a vertical or horizontal position to the number of blob pixels in the horizontal or vertical direction correspondingly at that position. In the literature [3], it is reported that the ratio of the second difference, $VP(x-1) - 2 * VP(x) + VP(x+1)$ to the value $VP(x)$ is useful for estimation of breakpoint of touching characters in normal text. We also have found that this criterion is valid for formulas.

Our criteria for the breakpoint are as follows.

(1) Horizontal segmentation

"In the middle part (80%) of the blob, the horizontal breakpoint is the position where the value of VP is lowest and the second difference is positive".

(2)Vertical segmentation

The vertical breakpoint can be estimated similarly. But in our observation of touching characters in formulas, vertical touching characters often occur when a horizontally elongated symbol (e.g., horizontal bar of a fraction or horizontal line segment of symbol $\sum$ etc.) and characters are printed closely. For these cases, it is also suitable to estimate the breakpoint by rapid change of the value HP. The criterion for the vertical breakpoint is the following.

Fig. 2. Horizontal and vertical projection profiles. The gray lines show estimated breakpoints

"In the middle part (80%) of the blob, the vertical breakpoint is the position where the ratio $HP(y)/HP(y+1)$ is less than θ_1 or greater than θ_2. If there is no such point, the breakpoint is estimated in the same way as in the case of the horizontal segmentation".

In the above criteria, the reason we examine only the some extent of middle part of a blob is to exclude the possibility that the blob may be segmented into extremely a large component and a small one.

The blob of touching characters in a text printed with the Roman font can be separated by a vertical line at a breakpoint. However in formulas, most characters are printed with italic font, therefore there are many cases touching characters can not be separated by the vertical line. A proper line which separate a merged blob into two components can be estimated by the valley between them corresponding to each character in the gray-scale image of the given blob. Fig.3 shows the blurred blob of Fig.2 obtained by applying the Gaussian kernel as a point spread function to the original binary image.

$$g(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

Some advantages in character segmentation problem using gray-scale image are stated in the literatures [4,5]. In the literature [4], small value of σ is used to approximate stroke lines of a character, but large one is used to outline its shape and position in the literature [5]. We also adopt a large value of σ which

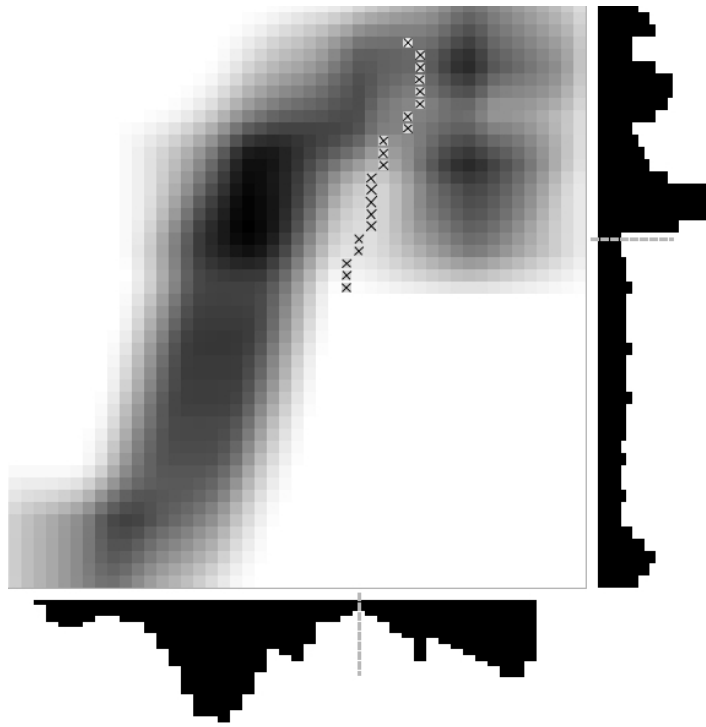


Fig. 3. Blurred blob of Fig.2. Pixels marked with "x" show horizontal minimal points

corresponds to the number of pixels larger than the width of strokes and smaller than the size of characters. In order to estimate the cutting line, we calculate the minimal points in the gray-scale image horizontally and vertically. In Fig.3, pixels marked with "x" show the horizontal minimal points. This figure shows that minimal points of the blurred blob locate in the valley between two characters and can be used to estimate a cutting line. Fig.4 shows the horizontal cutting line at the breakpoint obtained by using the method of least square from the minimal points near it.

The both of components segmented by the cutting lines are fed to the classifier and their average scores are calculated in horizontal and vertical segmentation. Then the segmentation having the higher average score is adopted.

3 Experiments and Remarks

The method described so far was implemented in the C Language and incorporated into our formula recognition system [2]. Some experiments were carried out for formulas scanned from some kinds of printed journals. Fig.5 shows the

Fig. 4. Estimation of a cutting line in horizontal segmentation

recognition results for inputed images which contain horizontal, vertical and diagonal touching characters. In our system, the decision whether a given blob should be segmented or not depends on the score of the first stage classifier for every connected components in formulas. In Fig.5, every score for blobs of touching characters is less than a predetermined value and segmentations for them were tried. Finally, each segmented component has higher score than the original blob, and all the formulas in Fig.5 are recognized correctly. Our method of segmentation is very simple and can be executed very fast. We are currently investigating how to segment touching characters where more than three characters are merged.

$$-1 \le \frac{\iint fg}{\sqrt{\iint f^2}\sqrt{\iint g^2}} \le 1$$

$$-1 \le \frac{\int\int fg}{\sqrt{\int\int f^2}\sqrt{\int\int g^2}} \le 1$$

$$\theta_i = \frac{2\pi i}{N}$$

$$\theta_i = \frac{2\pi i}{N}$$

$$\Delta x_i' = \Delta x_i + \frac{\partial u_i}{\partial x_j}\Delta x_j$$

$$\Delta x_i' = \Delta x_i + \frac{\partial u_i}{\partial x_j}\Delta x_j$$

$$-\sum_{i\in I_k}\left[\sum_{j=1}^{n} w_{ij}z_j + \theta_i\right]\Delta x_i(t)$$

$$-\sum_{i\in I_k}\left[\sum_{j=1}^{n} w_{ij}z_j + \theta_1\right]\Delta x_i(t)$$

Scanned images

Recognition results

Fig. 5. Experimental results

References

1. Okamoto, M., Miao, B.: Recognition of Mathematical Expressions by Using Layout Structure of Symbols. Proc. of ICDAR'91 (1991) 242-250
2. Okamoto, M., Higashi, H.: Structure Analysis and Recognition of Mathematical Expressions. Proc. of ICDAR'95 (1995) 430-437
3. Kahan, S., Pavlidis, T., Baird, H. S.: On the Recognition of Printed Characters of Any Font and Size. IEEE Trans. of PAMI, **9**, 2 (1987)
4. Wang, L., Pavlidis, T.: Direct Gray-Scale Extraction of Feature for Character Recognition. IEEE Trans. of PAMI, **15**, 10 (1993) 1053-1067
5. Nako, K., Takamatsu, R., Sato, M., Kawarada, H.: A New Character Segmentation Method for Handwritten Documents Based on Differential Structure of Blurred Document Image. Technical Report of IEICE, PRU93-131 (1994) 9-16