Multi-Window Binarization of Camera Image for Document Recognition

In-Jung Kim
Inzisoft, co., ltd.
ijkim@inzisoft.com

Abstract

This paper proposes an adaptive binarization method for the document image acquired by camera. By applying multiple windows vary in size, it determines the local thresholds with the information from the global trend as well as the local detail. As the consequence, the proposed method is good at not only preserving fine detail of the character structure but also absorbing noise. The effectiveness of the proposed method was shown by the experiment.
Keywords: adaptive binarization, camera image

1. Introduction

The document recognition technology has been successively applied to many practical systems. Most of the recognition systems were developed for the document image acquired by scanner. However, as various kinds of camera are getting popular, the need for the recognition of the camera image is getting more important.

The camera image is more difficult to recognize than the scanner image, because with a camera, it is difficult to control the imaging environment as much as we can with the scanner. Even if the user is assumed to take the image carefully, the camera image often has the following problems: Firstly, the brightness is not uniform because of uneven lighting or aberration of the camera lens. Secondly, the gray level surface of the camera image is smoother than that of the scanner image. In other words, the edge of the character is not as clear as that of the scanner image, and therefore, the difference in intensity between the foreground and the background is obscure in many camera images. These problems are especially significant for the images that are not focused well.

In most of the recognition system, the document analyzing procedures are followed by a binarization process. In order to deal with the camera image, performance of the binarization algorithm is very important. However, the performances of the conventional local binarization methods depend upon the parameters. Even though a set of parameters works nicely for an image, we can't believe those parameters would be suitable for other images. Moreover, even when we provide an optimal parameter carefully, the binarization algorithm can fail to preserve important details of the character structure. Consequently, there are still remained problems in binarization of the camera image.

This paper proposes a new binarization method to alleviate these problems. The proposed method applies multiple windows vary in size to decide the local thresholds. Each window reflects the information from different range from the target pixel. The sizes of the windows are automatically determined through an analysis of the document image, and therefore, it is not sensitive to the variation in property of the document image.

The rest of this paper is organized as follows: Section 2 introduces the previous works as well as an analysis of their problems. Section 3 and 4 describes the proposed method and their experiment results. Finally, the conclusion was provided in section 5.

2. Related works

2.1. Previous works

Many techniques have been developed to binarize the gray image. For the global binarization, the most important is determining the global threshold. The most popular algorithms are Otsu's method, which chooses the threshold that minimizes within-group variance, and Kittler and Illingworth's method, which chooses the threshold that minimizes Kullback information distance [1][2]. Huang and Wang's fuzzy binarization method is also widely used [3].

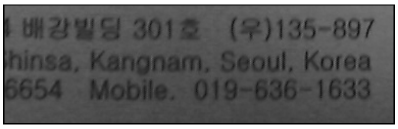
On the other hand, many popular local binarization methods are window-based approaches, in which, the local threshold for a pixel (x, y) is computed from the gray values of the pixels in the window centered at (x, y). Many researchers proposed various formulas to compute the local threshold. For example, Bernsen computed the local threshold from the minimum and the maximum gray values in each window, while Niblack computed it from the mean and the standard deviation [4][5]. Eikvil, et. al. computed the local threshold by applying Otsu's method in each window [6]. Nakagawa and Rosenfeld computed it by analyzing the bimodality of the gray values in each window [7]. Trier evaluated various binarization methods from the view point of the recognition performance [8]. He reported that Niblack's method provided the best performance in his experiment

However, Sauvola, et. al. pointed out that Niblack's method does not work well for cases in which the background contains light texture[9]. In order to alleviate this problem, he modified the binarization formula as formula (1), where R is the dynamic range of standard deviation. He also argued that his method is not very sensitive to the parameter k.

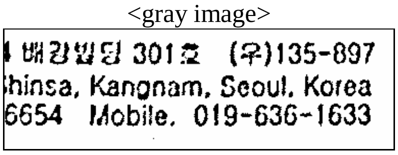
$$T(x,y) = m(x,y)[1 + k(\frac{s(x,y)}{R} - 1)] \quad (1)$$

2.2. Analysis of window-based method

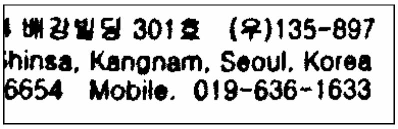
Applying Sauvola's method to various camera images, we found the performance of Sauvola's algorithm depends on the window size. Especially, the suitable window size for a document image was heavily affected by the character size as well as the character thickness. If the window size is too large, the details of the character structure were degraded in many cases. In figure 1(a), the first image contains small characters and the difference between the background and the foreground is not very clear.



<gray image>

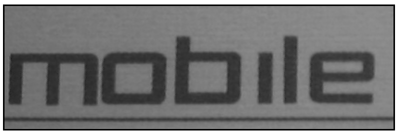


<window size = 12>



<window size = 30>

(a) small font



<gray image>

<window size = 12>

<window size = 30>

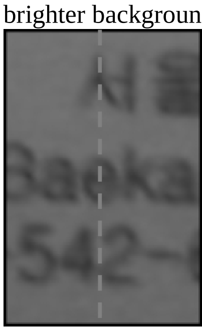
(b) thick font

Fig. 1: Binarization results by Sauvola's method.

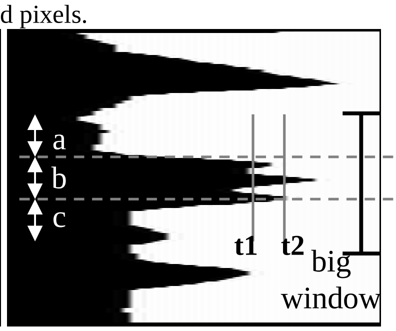
Although it is not easy to separate the character region from the background clearly in such an image, Sauvola's method produced a reasonable result as the second image of figure 1(a). In that case, the optimal window size 12 was determined manually. However, with a larger window, for example, 30, the details of the character structure were degraded as the last image of figure 1(a).

On the other hand, if the window size is too small, the central areas of thick and black regions were often classified into the background as the second image of figure 1(b). This problem could be avoided by using larger window, which produced the problematic result in figure 1(a).

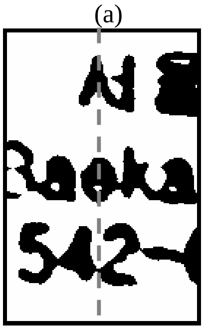
Figure 1 clearly shows that no meter what value we assign, a single fixed value for the window size cannot cover all kind of document images. The reason can be explained by figure 2, 3, and 4. In the camera image, the intensity of a pixel is significantly affected by the neighbors. Therefore, the background pixels nearby a foreground region are usually darker than other background pixels, as figure 2(a) and 2(b). Figure 2(b) shows the gray values along the dotted line in figure 2(a). In order to separate the hole of the character 'e' from its foreground neighbors, the local thresholds in that region should be between ***t1*** and ***t2***. However, if the window is too large, the local threshold is likely to be assigned a smaller value than ***t1*** affected by the background pixels in area ***a*** and ***c***, which are much brighter than the hole. To avoid this problem, the window size should be small enough to reduce the effect of the brighter background pixels.



(a)



(b)



(c)

Fig. 2: Big window problem.

On the other hand, the second problem can be explained by figure 3. If the window size is too small, compared with the character thickness, the window centered at an inner pixel produces a large mean and a small standard deviation, because most pixels in that window belong to the foreground. As the consequence, the local threshold for such a pixel is likely to be assigned a large value, and therefore, the corresponding pixels can be classified into the background.

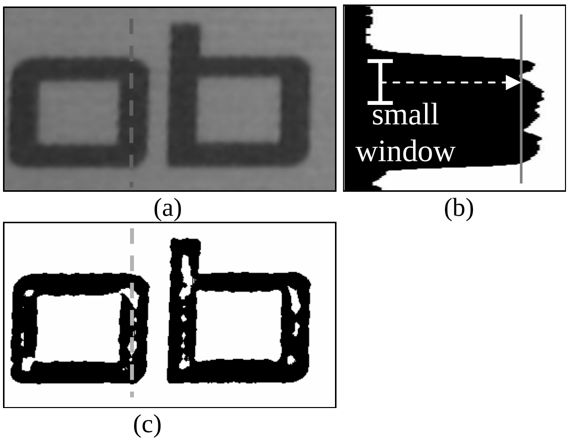


Fig. 3: Small window problem 1.

There is another problem with a small window as shown in figure 4. The local binarization algorithm can produce a noisy result with a small window, because the small peaks in a rough background region can be classified into the foreground regions. In a point of view, we can say this problem is from the basic idea of the local binarization, because the pixels in such a peak are darker than their neighbors.

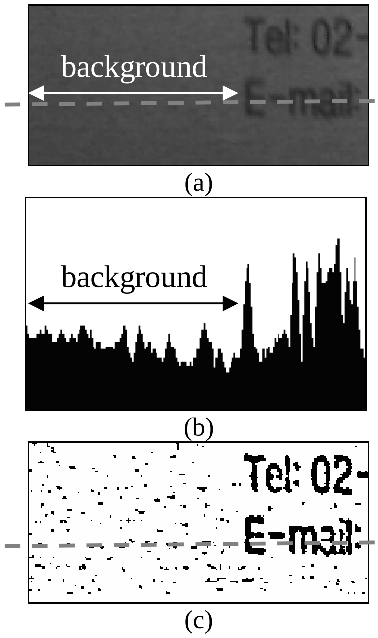


Fig. 4: Small window problem 2.

The reason of the first problem is that the fine detail of the character structure is often ignored with the large window. On the other hand, the second and the third problems are because the small window cannot reflect the global trend. These problems are the inherent of the algorithms that are based on a single fixed-size window.

3. Adaptive binarization using multiple windows.

In order to avoid the above problems, the binarization algorithm should determine the local thresholds considering not only global trend but also local detail. For this reason, we propose a new binarization method that applies three windows. The first window is used for a preliminary binarization for the analysis of the document image. The second and the third windows are used to determine the actual local thresholds. Each of the two windows reflects the global trend and the local detail, respectively.

3.1. Framework

The overall flow of the proposed algorithm is as follows: Firstly, the input image is binarized with a moderate-size window. With the preliminary binarization result, it extracts text lines and analyzes the property of each text line. Then, for each text line, it decides the sizes of the second and the third windows according to the analysis result. The size of the second window is assigned a large value enough to reflect the global trend. On the other hand, the size of the third window is assigned a small value enough to catch on the detail of the character structure. And finally, it binarizes again each text line with both of the second and the third windows to make the final result.

3.2. Text line extraction

We applied a simple algorithm to extract text lines. Firstly, the connected components are extracted from the preliminary binarized image. Then, it iteratively merges the pair of connected components that are vertically overlapped. When no pair of connected components can be merged any more, the iteration terminates.

3.3. Determination of window sizes

The size of the second window should be large enough to avoid the second and the third problems described in section 2.2. From figure 3 and 4, we can claim that those problems can be avoided with a window whose size is larger than the character size. If the window size is larger than the character size, it usually includes sufficient numbers of background pixels as well as foreground pixels, regardless of its position, and as the result, it doesn't generate the biased threshold. Therefore, the proposed algorithm chooses the average height of the characters for the size of the second window.

On the other hand, the size of the third window should be small enough to avoid the first problem described in section 2.2. From figure 2, we can also claim that problem can be avoided with the window whose size is comparable to the character thickness. With such a window size, the bright background pixels are rarely included in the window centered at a dark background pixel. As the result, the dark background can be separated from its foreground neighbors.

The character thickness can be estimated by analyzing run-length histogram. Run is a group of horizontally consecutive black pixels. Firstly, all runs in each text line region are extracted by a horizontal scan. Then, a run-length histogram is made by counting the runs according to their length. From the run-length histogram, the length with the largest frequency is the estimation of the character thickness.

Consequently, the sizes of the large window and the small window are same with the character height and the character thickness, respectively. In order to apply the proposed approach, we need to assume the size and the thickness of the characters do not vary very much. Fortunately, such an assumption is satisfied for each text line in most document images.

3.4. Binarization with multiple window

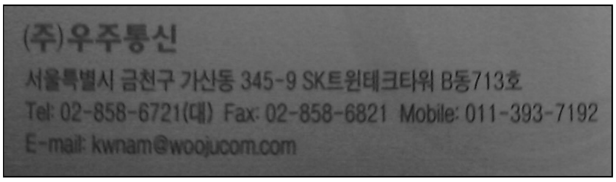
If the size a window is determined, the local threshold can be computed by formula (1). Since the proposed method employs two windows, it generates two local thresholds $T_{large}(x, y)$ and $T_{small}(x, y)$ for every pixel (x, y), each of which is computed with the large window and the small window. Conceptually, $T_{large}(x, y)$ was determined considering the global trend while $T_{small}(x, y)$ was determined considering the local detail. Firstly, all pixels are binarized with the threshold from the large window. Then, the resulting image does not suffer from the small window problems explained in section 2.2, because the size of large window was assigned a large value enough to avoid the small window problem. However, the fine detail of the character structure can be degraded in such an image. Therefore, the foreground pixels in the intermediate result are binarized again with the integrated threshold computed from the weighted average of the two thresholds as formula (2)

$$T(x, y) = \alpha T_{large}(x, y) + (1-\alpha)T_{small}(x, y) \qquad (2)$$

, where α is a coefficient that controls the effect of the global and the local information. From our experiment, the performance of the proposed method was promising with 0.1 ~ 0.3 for α, and the performance of the proposed method is not very sensitive to α.

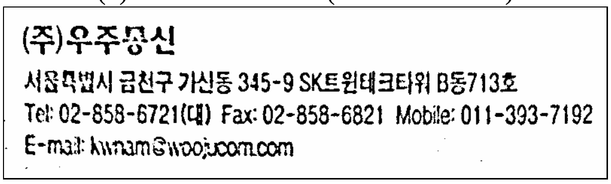
4. Experiments

4.1. Binarization results

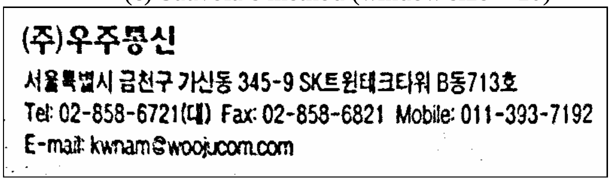


(a) gray image

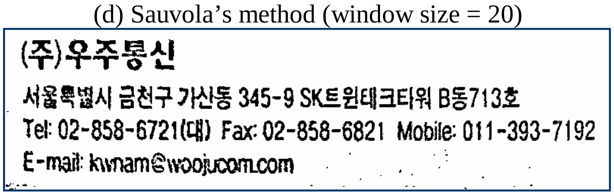
(b) Sauvola's method (window size = 5)



(c) Sauvola's method (window size = 10)



(d) Sauvola's method (window size = 20)



(f) Multi-window binarization

Fig. 5: Binarization results.

Figure 5 shows an example of the binarization result. In figure 5(a), the difference in gray level of the foreground and the background is not clear. Figure 5(b), (c) and (d) show the result of Sauvola's method with different window sizes. When window size was small as (b), the result image was noisy although the fine detail of the character structure was well preserved. On the other hand, when the window size was large as (d), the details of the characters were degraded. We manually found 10 for the window size provided the best result as (c). In every case, the constant factor *k* was manually determined to make the best result.

On the other hand, the result of the proposed method was presented in figure 5(f), which shows even better result than (c), the best result of Sauvola's method with the manually optimized parameters. In figure 5(f), the fine details of the character structure were quite well preserved, while most of the small peaks in the rough background were ignored.

4.2. Recognition test

In order to confirm the effectiveness of the proposed method in document recognition, we evaluated it with the recognition performance. For the evaluation, we collected a database of business card images taken by several kinds of million-pixel camera phones. The database was categorized into two sets. One contains well-focused 141 images, while the other contains 72 images not focused very well. For the recognition, we applied the Hangul recognizer and the numeral recognizer of *Inzi i-Form*^{TM}, a commercial document processing tool kit [10]. The target fields were Hangul name field and phone number field. The recognition rates are presented in table 1.

	Sauvola's method		Proposed method	
	Hangul Name	Phone Number	Hangul Name	Phone Number
Set 1	96.15 %	98.79 %	96.15 %	99.35 %
Set 2	57.14 %	95.00 %	85.71 %	96.71%

Table 1: Recognition rates.

For the good images in set 1, the proposed method was better than Sauvola's method, but the difference was not very big. However, for the bad images in set 2, the difference was significant.

5. Conclusion

As the camera is getting more popular, the need for the recognition of the camera image is getting more important. However, conventional binarization methods are not sufficient to build a high-performance recognition system for the camera image. In this paper, we presented an analysis of the problems of the conventional window-based local binarization methods, as well as a new adaptive binarization method to avoid those problems.

The proposed method determines the local thresholds considering not only the global trend but also the local details, by applying multiple windows vary in sizes. Moreover, the proposed method is highly adaptive to various document images, because it determines the window sizes according to the size and the thickness of the character. As the consequence, the proposed method is good at not only preserving the fine detail of the character structure, but also in absorbing the noise.

From the experiment, the proposed method produced better results than Sauvola's method. The recognition test also showed the proposed method is effective for the document image recognition acquired by the camera.

Reference

[1] N. Otsu, A threshold selection method from grey level histogram, IEEE Trans. SMC-9, pp. 62-66, 1979

[2] J. Kittler and J. Illingworth, Minimum error thresholding, Pattern Recognition 19, 41-47, 1986

[3] L. K Huang and M. J. Wang, Image thresholding by minimizing the measure of fuzziness, Pattern Recognition, 28, pp. 41-51, 1995

[4] J. Bernsen, Dynamic thresholding of grey-level images, Proc. 8th ICPR, pp.1,251-255, Paris, 1986

[5] W. Niblack, An Introduction to Image Processing, Prentice-Hall, Englewood Cliffs, NJ, 1986, pp.115-116

[6] L. Eikvil, T. Taxt, and K. Moen, A fast adaptive method for binarization of document images, Proc. 1st ICDAR, pp. 435-443, Saint-Malo, France, 1991

[7] Y. Nakagawa and A. Rosenfeld, Some experiments on variable thresholding, Pattern Recognition, Vol. 11, No. 3, pp. 191-204, 1979.

[8] O. D. Trier, Goal-directed evaluation of binarization methods, IEEE PAMI Vol. 17, No. 12, 1995.

[9] J. Sauvola and M Pietikainen, Adaptive document image binarization, Pattern Recognition, Vol. 33, pp. 225-236, 2000.

[10] Homepage of Inzisoft, co., ltd., http://www.inzisoft.com

0-7695-2187-8/04 $20.00 © 2004 **IEEE**