

Adaptive Thresholding Technique for Binarization of License Plate Images

Min-Ki Kim*

*Education Research Institute, Department of Computer Science Education,
Gyeongsang National University, Jinju 660-701, Korea*

(Received October 15, 2010 : revised December 7, 2010 : accepted December 8, 2010)

Unlike document images, license plate images are mostly captured under uneven lighting conditions. In particular, a shadowed region has sharp intensity variation and sometimes that region has very high intensity by reflected light. This paper presents a new technique for thresholding license plate images. This approach consists of three parts. In the first part, it performs a rough thresholding and classifies the type of license plate to adjust some parameters optimally. Next, it identifies a shadow type and binarizes license plate images by adjusting the window size and location according to the shadow type. And finally, post-processing based on the cluster analysis is performed. Experimental results show that the proposed method outperformed five well-known methods.

Keywords : Adaptive threshold decision, Shadowed license plate image, Cluster analysis
OCIS codes : (100.2000) Digital image processing; (070.4560) Data processing by optical means;
(110.2960) Image analysis; (070.5010) Pattern recognition and feature extraction

I. INTRODUCTION

As the number of vehicles increases rapidly, vehicle related problems such as traffic congestion at a toll gate, violation of traffic signals, and illegal parking get worse. An intelligent transportation system (ITS) has attracted attention as the most effective solution to these problems. Automatic identification of a vehicle is an essential element for developing ITS. The license plate recognition (LPR) by optical character recognition has been an active research area for the last ten years. A typical LPR system can be roughly divided into three modules: license plate detection, character segmentation, and optical character recognition. Some studies are based on gray-level pattern matching, which segments and recognizes the characters in a license plate without thresholding [1, 2]. However, most of the recent studies have the pre-processing step of thresholding before the character segmentation. The thresholding makes it possible to use various techniques developed in the field of document recognition.

The thresholding techniques can be roughly categorized as global methods [3-5] and local methods [6-8]. While the global methods select a single threshold value to classify an image into object or background classes, the local methods calculate an adaptive threshold value in each window or block.

The method of Yang et al. [5] is based on the assumption that the ratio of the character region to a license plate area is approximately fixed. In fact, the ratio varies a little according to the characters in the plate. Yang et al. [4] has proposed a method of adjusting the threshold according to the ratio of the character region which is obtained by an iterative thresholding. These methods are simple and effective for the license plate images which are captured from restricted places such as a toll gate or a parking lot entrance where the illumination can be controlled artificially. If we use these methods under unrestricted outdoor conditions, we cannot expect good thresholding results.

A local thresholding method is required to adjust to a local intensity variation. Bernsen [6] and Niblack [7] are well-known classical methods of local thresholding. Niblack's method computes a local threshold from the mean and the standard deviation of the gray values in a window, and Bernsen's method uses the minimum and the maximum of the gray values. So, Bernsen's method is more sensitive to the local variation than Niblack's. To binarize a license plate area, Yang et al. [9] tested some classical thresholding methods and selected Bernsen's method. Three classical thresholding methods were used for verification of a license plate in Tan and Chen [10]. First, they used Otsu's

*Corresponding author: mkkim@gnu.ac.kr

Color versions of one or more of the figures in this paper are available online.

method [3]. If the binarization result is not good enough for extracting desired features, then Bernsen's and Niblack's methods are used sequentially. Wu et al. [11] selectively adopted two different thresholding methods according to the ratio of edge pixels in a window. These local methods show good results in normal outdoor illumination conditions, but they had shown broken strokes and ghost objects at the boundary where intensity changes drastically. Since sharp intensity variation is not unusual in a rear license plate, a new thresholding method which is robust to the drastic intensity variation is necessary to develop a practical LPR system. Although a variety of methods have been proposed for developing an LPR system, they did not consider cast shadows on license plates. Thus, the problem of the sharp intensity variation from the cast shadows has rarely been studied. This paper proposes a robust thresholding technique for license plate images with shadowed regions. First, it performs a rough thresholding and classifies the type of license plate to adjust some parameters optimally. Next, it identifies a shadow type and binarizes the license plate image by adjusting the window size and location according to the shadow type. Finally, post-processing based on cluster analysis is performed.

The rest of this paper is organized as follows. Section 2 introduces some related work and section 3 presents a detailed description of the proposed method. In section 4, experimental results and analysis are provided, and finally the conclusions are described in section 5.

II. RELATED WORK

Although a large number of studies have been made on license plate localization and recognition, only a few studies use a thresholding method for shadowed license plate images. In this section, we review some existing methods. The first three methods have been used or proposed to binarize license plate images. The last two methods have been proposed to binarize uneven lighting and degraded document images. Although the last two studies deal with document images, not license plate images, these methods are also reviewed because they deal with uneven illumination problems.

Niblack's method is based on the calculation of the local mean and local standard deviation [7]. The threshold of a pixel at (x, y) is computed by Eq. (1), where $m(x, y)$ and $s(x, y)$ are the average and standard deviation of gray values of the pixels in the window at (x, y) . The size of the window must be small enough to reflect local details, but at the same time large enough to suppress noise. The suitable window size for a document image was heavily affected by the character size as well as the character thickness [12]. The value of k is used to adjust the ratio of total pixels that belong to the foreground object especially for the boundaries of the object. It has been known that a value of $k=0.2$ separates objects well enough from a

background when the objects are darker than the background

$$T(x, y) = m(x, y) + k \cdot s(x, y) \quad (1)$$

This method does not work well when the background area contains local variations due to stains or uneven illumination. To solve this problem, Sauvola and Pietikainen [8] have proposed a modified Niblack method. The thresholds are computed with the dynamic range of standard deviation, R . Furthermore, the local mean is utilized to amplify the contribution of standard deviation in an adaptive manner. This efficiently removes the noise due to the local variation in the background. In their experiments with 8-bit gray level images, $R=128$ and $k=0.5$ showed good results.

$$T(x, y) = m(x, y) \cdot \left[1 - k \cdot \left(1 - \frac{s(x, y)}{R} \right) \right] \quad (2)$$

Wu et al. [11] have proposed an alternative thresholding method based on the boundary characteristics. This study uses the boundary features to choose either the basic adaptive thresholding (BAT) method or c-means algorithm. The BAT is appropriate for segmenting two clusters whose variances and possession rates are similar. In the opposite case, the c-means algorithm is appropriate. At first, an image is partitioned into $M \times N$ block images. The gray value of each pixel in the (i, j) th block image is denoted as $f_{i,j}(x, y)$. The set of boundary pixels in $f_{i,j}(x, y)$ is defined as follows, where the T_d is a predefined threshold value.

$$A_{i,j} = \{(x, y) \mid |f_{i,j}(x, y) - f_{i,j}(x-1, y)| \geq T_d\} \quad (3)$$

After the set of boundary pixels is obtained, the updated threshold T_u is set as the average of gray-levels of the boundary pixels. For the threshold of the (i, j) th block image, the updated probability ($P_{u,0}$) is defined as follows:

$$P_{u,0} = \frac{\sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \begin{cases} 1 & \text{if } f_{i,j}(x, y) < T_u \\ 0 & \text{otherwise} \end{cases}}{M \times N} \quad (4)$$

$P_{u,1}$ represents the ratio of pixels with higher gray value than T_u , and it is computed by $1 - P_{u,0}$. If the difference of $P_{u,0}$ and $P_{u,1}$ is less than a predefined threshold T_p , then the corresponding block is binarized by BAT method. Otherwise, the block is binarized by the c-means algorithm. It uses $T_d=8$ and $T_p=0.1$. This method is faster than pixel-based methods [7, 8], but it may have unexpected ghost objects.

In order to binarize uneven lighting images, Huang et al. [13] partitioned the whole image into small blocks that did not overlap, and then used existing methods to binarize each block. The technique is based on a pyramid data structure, and the block size is adaptively selected according to the Lorentz information measure (LIM). The LIM indicates whether each block contains both the object

and the background. If the block contains the object and the background, Otsu's method [3] can be directly applied for thresholding the block. Otherwise, adjust the block size continuously according to the pyramid data structure until it meets the requirement.

A document can be degraded by non-uniform illumination, low contrast, large signal-dependent noise, smear and strain. Gatos et al. [14] have tried to solve these problems by a background surface thresholding. In order to estimate foreground regions, it first gets an initial binary image $S(x, y)$ by using Sauvola's method [8]. Then, it computes an approximate background surface $B(x, y)$ of the input gray image. The background surface estimation is guided by the valuation of the $S(x, y)$ image. For pixels that correspond to the background at the $S(x, y)$ image, the corresponding value at $B(x, y)$ is equal to $I(x, y)$. For the remaining pixels, the value of $B(x, y)$ is computed by a neighbouring pixel interpolation. The final binary image $T(x, y)$ is given by the following formula:

$$T(x, y) = \begin{cases} 1 & \text{if } B(x, y) - I(x, y) > d(B(x, y)) \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where $d(B(x, y))$ is a threshold that adaptively changes according to the gray value of the background surface $B(x, y)$ in order to preserve textual information even in very dark background areas. After the final binarization, post-processing is performed to eliminate noise, improve the quality of text regions and preserve stroke connectivity.

III. THE PROPOSED METHODS

This section presents a detailed description of the proposed method for thresholding license plate images. The overview of our method is illustrated in Fig. 1. First, it makes a rough binary image (RBI) by using Niblack's method [7]. Next, it generates a binary image using both a gray image and the rough cluster information. The type information of a license plate and the rough cluster information are used for the second thresholding. At last, a final binary image is acquired by post-processing based on cluster analysis. Not only the binary image but also its gray image is used for the post-processing.

3.1. Classification of a License Plate Type

When an adaptive thresholding method is used, the

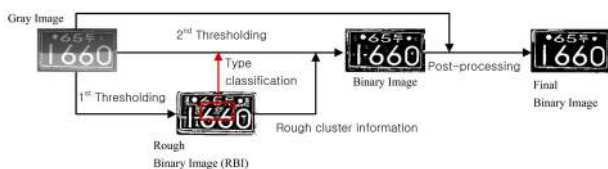


FIG. 1. Overview of the proposed methodology.

quality of a binarized image is highly affected by the parameters of window size and k value [15, 16]. Thus, it is important to find the suitable window size and k value for target images. The sign of k is especially important in a license plate image, because it has polarity. Though there are various kinds of license plates that have different combinations of character and plate colors, the license plates can be categorized into two types if we classify them by the viewpoint of binarization. One is white-character type (WCT) of which background color is darker than the character's color. The other is a black-character type (BCT) of which background color is lighter than the character's color. The top left image in Fig. 2-(a) is an example of WCT image, and the bottom left image is an example of BCT image. When using the method of Niblack, as shown in Fig. 2, the plus sign of k is suitable for WCT images, and the minus sign of k is suitable for BCT images.

In order to set the sign of k correctly, we must know the type of license plate before binarization. We propose a simple and effective algorithm for classifying the type of license plate. First, a rough binary image (RBI) is obtained by using Niblack's method with the window size of 11×11 and $k=0.0$. Then, the central region of RBI is examined and the number of white pixels (N_w) and the number of black pixels (N_b) are counted. The width and height of the central region are equal to 40% of the width and height of the image. The license plate type is determined by the ratio of the white pixels and black pixels in the central region. If N_w is greater than N_b , then the license plate image is categorized into the BCT. Otherwise, the ratio (Rd) calculated by the following equation is tested.

$$R_d = (N_b - N_w) / (N_b + N_w) \quad (6)$$

If R_d is greater than or equal to 0.2, the license plate image is categorized into the WCT, otherwise its type is determined as BCT. Managing two different kinds of binary images is inconvenient for the succeeding processes. So, in the case of a BCT image, the binarized result is reversed for a unified representation where the character and background are displayed as white and black respectively.

3.2. Shadow Detection and Window Adjustment

We know that the character size and thickness in a document are important factors to decide the suitable window size in

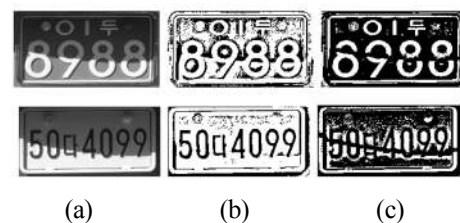


FIG. 2. The polarity of license plate images: (a) gray images; (b) and (c) binarized results with $k=-0.5$, $k=+0.5$ respectively.

the window-based adaptive method [12]. It does not mean that all the characters in a license plate have the same size and thickness, but their size and stroke width should be proportional to the size of the license plate. Thus, it is natural to adaptively change the window size according to the size of a license plate image. As far as the window size is greater than stroke width, the quality of binarized image has not been highly affected by the character size. However, the quality has been severely affected by the window size when intensity changes drastically by cast shadows. Fig. 3 shows two examples of license plate images with a cast shadow and their results of thresholding with different window size. The size of the window is defined by $(2l+1) \times (2l+1)$, where l is the length from the central point of its window to the window boundary. In Fig. 3, from left to right, width and height of each window is $H/3$, $H/7$, $H/11$, and $H/15$ respectively, where H is the height of a license plate image. As expected, the smallest window provides locally adapted thresholds, so broken strokes and ghost objects appear only on the border of the cast shadow but many salt and pepper noises occur. On the contrary, the larger window makes the problems of broken strokes and ghost objects more severe than the smaller window.

To acquire good binarization results, the problems of the broken strokes and the ghost objects should be solved. This paper proposes a new method to solve these problems. The basic idea is, first, to detect the boundary of a cast shadow, and then to adjust the window size and location according to the distribution of gray values in the window. There are various shapes of shadows, but we treat every shadow either as a horizontal shadow or a vertical shadow. We have defined six regions in a window as depicted in Fig. 4. The top region (R_{h1}) and the bottom region (R_{h3}) in Fig. 4-(a) and the left region (R_{v1}) and the right region

(R_{v3}) in Fig. 4-(b) have been defined to detect a horizontal shadow and a vertical shadow, respectively. The value of d in Fig. 4 has been set as $\max(3, l/5)$, where l is the length from the central point to the boundary of its window.

If the ratio R_{diff} computed from Eq. (7) is smaller than 0.5, we decide that a shadow exists in the window. In this case, if $|M_{h1} - M_{h3}|$ is greater than $|M_{v1} - M_{v3}|$, the shadow is classified as a horizontal shadow. Otherwise it is classified as a vertical shadow. $|M_{h1} - M_{h3}|$ and $|M_{v1} - M_{v3}|$ are the differences of average intensity values in the horizontal region R_{h1} and R_{h3} and in the vertical region R_{v1} , and R_{v3} , respectively. Not only a gray image but also its RBI is used to calculate the average intensity of each region. The difference of means between two regions, R_i and R_j , is computed as the weighted sum of the difference of the means in the foreground region and the difference of the means of the background region as described in Eq. (8), where FM_i and BM_i are the means of gray values of foreground pixels and background pixels in region R_i . The white pixels and the black pixels in the RBI belong to the foreground and background respectively. The foreground pixels in the RBI are less reliable than the background pixels. In addition, the difference of means in foreground regions is larger than that in background regions, so we have set the weight differently. In this study, the foreground weight w_f is 0.2 and the background weight w_b is 1.0. All the threshold values in Eq. (7) and (8) are acquired by experiments with some data. We found the proper values by comparing decisions made by our algorithm to those made by a human.

$$R_{diff} = \begin{cases} \frac{\min(|M_{h1} - M_{h3}|, |M_{v1} - M_{v3}|)}{\max(|M_{h1} - M_{h3}|, |M_{v1} - M_{v3}|)} & \text{if } \max(|M_{h1} - M_{h3}|, |M_{v1} - M_{v3}|) > Th_d \\ 1 & \text{otherwise} \end{cases} \quad (7)$$

$$|M_i - M_j| = |FM_i - FM_j| \times w_f + |BM_i - BM_j| \times w_b \quad (8)$$

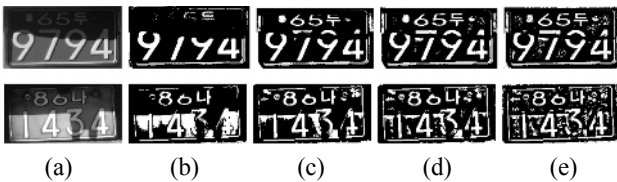


FIG. 3. Input image (a) and the thresholding results with the different window size $(2l+1) \times (2l+1)$: (b) $l=H/3$; (c) $l=H/7$; (d) $l=H/11$; (e) $l=H/15$.

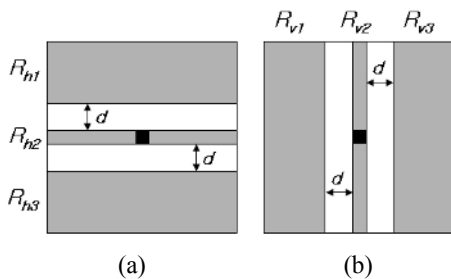


FIG. 4. Six regions to detect a shadow type.

The central horizontal region (R_{h2}) and central vertical region (R_{v2}) have been defined to detect the homogeneity between the central pixel of a window and a neighbouring region in the window. If a horizontal shadow is detected, the mean of R_{h2} is compared with the means of R_{h1} and R_{h3} . If $|M_{h2} - M_{h1}|$ is less than $|M_{h2} - M_{h3}|$, we use the homogeneous region R_{h1} as the window for applying Niblack's method. Otherwise, the region R_{h3} is used as the window. A similar approach is also applied to a vertical shadow. When thresholding the pixels at the boundary of a shadow, the adaptive window approach can fairly improve the problem of broken strokes. In Fig. 5-(a) and (c), the circles represent the location of pixels to be binarized and the rectangles represent their windows. The binarized image by Niblack's method is shown in Fig. 5-(b), where the second and fourth pixels, from left to right, at the boundary of the shadow are incorrectly binarized. Fig. 5-(d) shows the binarized image by the adaptive window, where only

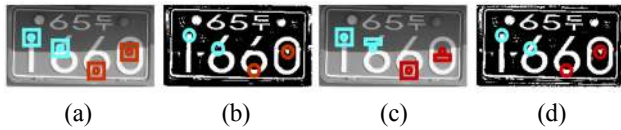


FIG. 5. Comparison of binarized results: (a) fixed windows; (b) binarized image by Niblack's method; (c) adaptive windows; (d) binarized image by the proposed method.

the fourth pixel is incorrectly binarized. The fourth pixel was not detected as a pixel located at the boundary of a shadow due to an error in the RBI. This kind of error will be corrected by the post-processing based on cluster analysis.

3.3. Post-processing Based on Cluster Analysis

In order to improve the quality of a binarized image, previous post-processing methods have usually used the information of the binary image only. Gatos et al [14] used a shrink filter to remove noise from the background and applied a swell filter to fill possible breaks, gaps or holes in the foreground. Yang & Yan [17] used run length features. These methods are effective at removing small noise such as salt and pepper noise, but their limitation is that ghost objects, as shown in Fig. 6, are not removed. This paper proposes a new post-processing method based on cluster analysis to remove the ghost objects. It uses not only the cluster information in the RBI but also the gray-level information in its original image. Binarization can be considered as categorizing all the pixels in the image into two clusters. If the resulting binary image is a reliable clustering result of its gray image, most of the pixels with similar gray values of a seed pixel may be categorized into the same cluster of the seed pixel. If the cluster of a seed pixel is not equal to the major cluster of the similar surrounding pixels, the cluster of the seed pixel is corrected by reversing.

The three pixels, denoted as $p1$, $p2$, and $p3$ in Fig. 6-(a), have been classified correctly, but the rightmost pixel, $p4$, has been incorrectly classified. Within the window $w1$, most of the pixels having the similar gray values to that of a seed pixel $p1$ have been classified into the foreground as shown in Fig. 6-(b). Thus, we can estimate that $p1$ is correctly binarized. The $p2$ and $p3$ are also similar cases. On the contrary, the majority of the pixels similar to $p4$ in the window $w4$ belong to the background cluster. In this case, the thresholding result of $p4$ is corrected by reversing.

A premise of the post-processing by cluster analysis states that the pixels in the same cluster may have similar gray values if the pixels are in the near distance. However, this premise is not satisfied when a cast shadow appears. Two different cluster analysis methods are applied sequentially. The first cluster analysis method (CAM_1) inspects only some pixels satisfying predefined conditions. In contrast, the second cluster analysis method (CAM_2) inspects all the pixels in a predefined window. CAM_1 probes 8-directional neighbouring pixels from a seed pixel. It continuously

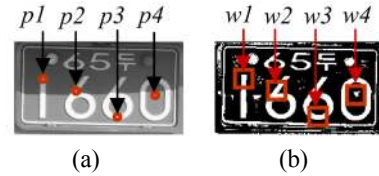


FIG. 6. An example of binarized result having ghost objects: (a) four seed pixels in a gray image; (b) four windows corresponding to each seed pixel.



FIG. 7. Post-processing results: (a) gray image; (b) binarized result before post-processing; (c) result of post-processing by CAM_1 ; (d) result of post-processing by CAM_2 .

examines the neighbour pixel in each direction and counts the number of hits and misses while the following condition is true.

$$|G(sx, sy) - G(nx, ny)| < Th,$$

where the $G(x, y)$ means the gray value of the pixel at (x, y) , the sx and sy are the coordinates of the seed pixel, the nx and ny are the coordinates of the neighbour pixel, and Th is a threshold for judging whether the two pixels are homogeneous or not. A hit means that a neighbouring pixel and a seed pixel are in the same cluster, and a miss means that they are in two different clusters. If the total number of misses is greater than the total number of hits, the seed pixel is corrected by reversing. Otherwise, the seed pixel is not changed.

After the post-processing based on CAM_1 , we then apply CAM_2 . CAM_2 uses the same approach as CAM_1 . It counts the number of hits and misses in the same way. However, it has two differences. The first is that it inspects all the pixels within the window that defines a probing region. The second is that it strengthens the criterion which is used to correct the seed pixel. When the hit ratio is less than a half of the miss ratio, the seed pixel is reversed. Normally, the window size of 11×11 is sufficient to correct a seed pixel by CAM_2 . Sometimes, an enlarged window is necessary for the seed pixel at the boundary of a shadow because there are many noises including ghost objects at the boundary. The window size is automatically adjusted according to the location of a seed pixel and the boundary of a shadow. Fig. 7 shows the results of post-processing by CAM_1 and CAM_2 .

IV. EXPERIMENTAL RESULTS

Two sets of data were used in this experiment. These data were obtained by segmenting the region of license

plate from the parked vehicle images of 1024×768 with 96 dpi. The average width and height of the license plate region is 217 and 105 pixels respectively, and the largest one is six times bigger than the smallest one. Set 1 consists of 120 license plate images captured under normal outdoor lighting conditions. Set 2 has 80 license plate images that intensity changes drastically at the boundaries of shadows.

In order to measure the performance of the proposed and comparison methods, we implemented them with C++ on a Pentium PC (2.2 GHz) and measured the quality of binarized results by using two evaluation criteria: misclassification error (ME) and relative foreground area error (RAE). The measure of misclassification error (ME) has been widely used to evaluate the performance of various thresholding methods [16, 18-19]. ME reflects the percentage of the incorrectly clustered pixels. It can be computed by the following equation:

$$ME = 1 - \frac{|B_r \cap B_t| + |F_r \cap F_t|}{|B_r| + |F_r|}, \quad (9)$$

where B_r and F_r denote the background and foreground of the reference (ground-truth) image, B_t and F_t denote the background and foreground area pixels of the test image.

RAE is a kind of measure to reflect the feature measurement accuracy that has been used under the name of relative ultimate measurement accuracy (RUMA) [16]. RUMA compares object properties such as area and shape, as obtained from the test image with the corresponding properties from the reference image. RAE measures the area feature by the following equation.

$$RAE = \begin{cases} (A_r - A_t) / A_r & \text{if } A_t < A_r \\ (A_t - A_r) / A_t & \text{otherwise,} \end{cases} \quad (10)$$

These two measures vary from 0 to 1, 0 for a perfectly binarized image, 1 for a totally incorrect result. When implementing these two measures, we do not compare the entire area of a license plate but compare only the inner area where characters are located. We have two reasons for that. The first is that the quality of the outer area is not important for license plate recognition. The second is that when we make reference images assigning some pixels in the outer region to foreground or background is more or less subjective. The white areas in the second and fourth images in Fig. 8 represent the inner area and the black areas represent the outer area. The inner area is defined by the region including the connected components of characters in a license plate image.

We compared the performance of the proposed method with the related work described in section 2: Niblack's method [7], Sauvola's method [8], Wu's method [11], Huang's method [13], and Gatos' method [14]. By using the method presented in section 3.1, we have perfectly classified the types of license plate images in set 1 and set



FIG. 8. Two reference images: Connected components of characters and their segmented inner area and outer area.

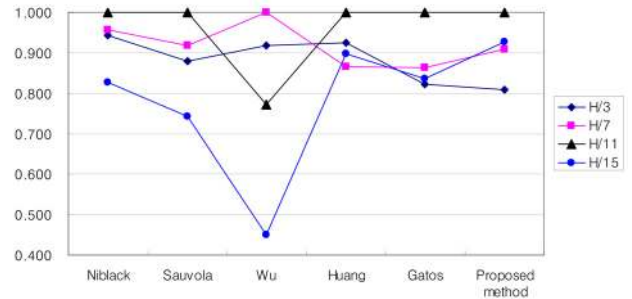


FIG. 9. Relative quality of binarization at each window size: $H/3$, $H/7$, $H/11$, and $H/15$.

2. The type of license plate is very useful information for a parameter setting. So, the information was equally used not only in the proposed method but also in the comparison methods. We first performed a parameter tuning for each method. Then, we measured the quality of thresholding result by using the measures of ME and RAE. We selected 40 images from set 1 for the parameter tuning. In Niblack's method, the experimental result shows the best quality at $k=\pm 0.5$ when we change k value from ± 0.1 to ± 0.5 . On the other hand, Sauvola's method shows the best quality at $k=\pm 0.2$. Four different sizes of window $(2l+1) \times (2l+1)$ were tested, where l is the length from the central point of its window to the outer boundary. Fig. 9 shows the experimental results of the 40 tuning data. The quality index at the vertical axis represents the relative quality of a specific window size to the optimal window size. The quality index (I_l) is computed by the following equations.

$$I_l = \frac{\min(ME_l + RAE_l)}{ME_l + RAE_l}, \quad (11)$$

where ME_l and RAE_l are the ME and RAE measured by using the window size of $(2l+1) \times (2l+1)$, and $\min(ME_l + RAE_l)$ means the minimum error by the optimal window size. I_l varies from 0 to 1. Fig. 9 shows the relative quality of thresholding at each window size, where the vertical axis represents I_l . If we categorize the six methods according to the scope of a single threshold, these methods can be divided into two groups: pixel-unit thresholding method (PUTM) and block-unit thresholding method (BUTM). The methods of Wu and Huang belong to BUTMs, and the rest belongs to PUTMs. In Fig. 9, all the methods except Wu's method show the best quality at the window size of $l=H/11$. Since Gatos' method is based on Sauvola's method, $k=\pm 0.2$ were used in Gatos' method. The proposed method set k value as ± 0.5 , because it is based on Niblack's method.

TABLE 1. Average processing time for binarization (unit: ms)

	H/3	H/7	H/11	H/15
Niblack	2500.5	598.2	304.8	207.0
Sauvola	2500.6	600.1	305.5	206.4
Wu	80.9	79.8	80.8	80.3
Huang	86.6	133.0	246.2	415.8
Gatos	2566.1	655.9	354.3	253.6
Proposed	2597.5	836.2	530.2	417.3

TABLE 2. Benchmark results of the proposed and comparison methods

	Set 1			Set 2		
	ME	RAE	Avg	ME	RAE	Avg
Niblack	.024	.041	.033	.037	.049	.043
Sauvola	.022	.053	.038	.055	.111	.083
Wu	.036	.088	.062	.114	.205	.159
Huang	.054	.104	.079	.259	.367	.313
Gatos	.032	.071	.051	.064	.127	.096
Proposed	.019	.042	.031	.026	.048	.037

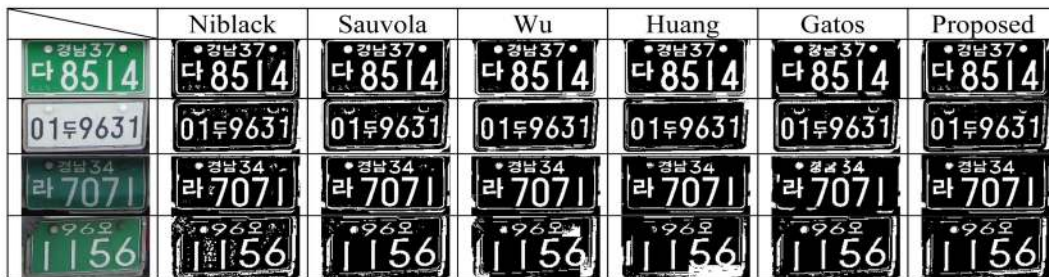


FIG. 10. Some sample images in set 1 and their visual benchmark results.

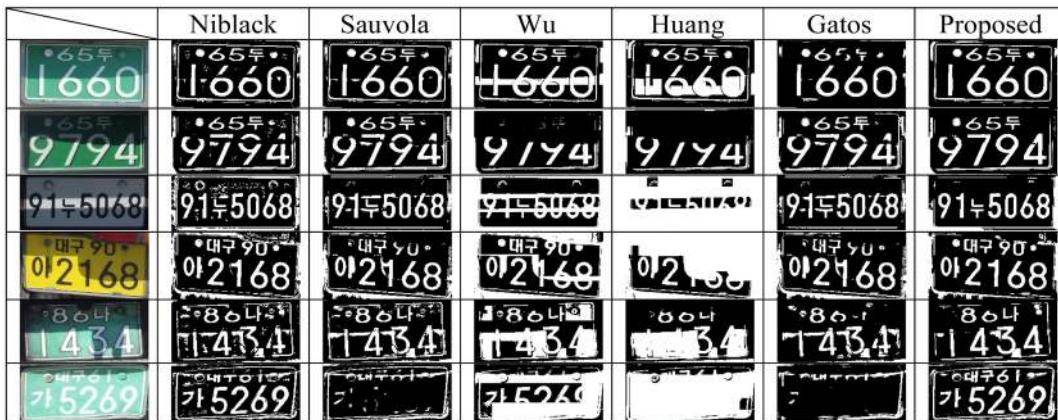


FIG. 11. Some sample images in set 2 and their visual benchmark results.

Table 1 shows the processing time of each method with different window size. On the whole, the BUTMs are much faster than the PUTMs, but the quality of binary image by the BUTMs is worse than for the PUTMs. PUTMs including the proposed method require a long processing time, but these methods are fast enough for application, that do not require real time processing, such as watching an illegally parked vehicle, finding a missing vehicle, etc.

After setting the parameters as optimal values from the tuning data, we benchmarked the quality of binary image generated by each method. All the images in set 1 and set 2 except the tuning data were used for benchmarking. The benchmark results are shown in Table 2. The PUTMs show better quality than the BUTMs. The performance of Sauvola's method was similar to that of Niblack's method

when we tested them with the images in set 1. In the case of set 2, however, Niblack's method showed much better quality than Sauvola's method. Sauvola's method adjusts the threshold at a bigger rate than Niblack's when the average intensity value is high. When a license plate image represents high intensity and variation, regardless of the existence of a shadow, Sauvola's method adjusts the threshold excessively. In that case, the quality of the binarized image is very poor. Especially when a strong shadow exists on a license plate, the quality goes from bad to worse. As shown in Table 2, the performance gap between the two methods is enlarged as the experimental data are changed over from Set 1 to Set 2.

The proposed method and Niblack's method showed a relatively low increase in ME and RAE as the test data

was switched from set 1 to set 2. That means that these two methods are robust to the illumination change. In contrast, the other methods showed a large increase in ME and RAE as the data is switched. The methods of Wu and Huang showed especially poor results in set 2.

By using the results presented in Table 2, we can rank the proposed and comparison methods, but we cannot compare the quality of binary image qualitatively. So, we observed the binary images generated by each method. Fig. 10 and Fig. 11 show some sample images and their binarized results. After a thorough observation of our experimental results, we obtained some important properties of each method. On the whole, all six binarization methods show good results with the images acquired from normal outdoor lighting conditions. In some images with uneven background, Wu's and Huang's methods tend to make ghost objects as shown in the last row of Fig. 10.

For set 2, each method shows various properties according to the input image. Niblack's method shows robustness to the drastic change of illumination, but the binarization results have some problems with broken strokes and ghost objects at the boundary of a shadow. Similar problems also appear in Sauvola's method. Sauvola's method shows an additional problem in the images with reflected light. The average intensity value of a region exposed to direct sunlight is near to 255 at an 8-bit gray image. So, the characters in the highlighted region disappeared as shown in the last row of Fig. 11. Gatos' method showed similar results. The methods of Wu and Huang are fragile to uneven lighting conditions, so they show very poor results in most of the images in set 2. Since Gatos' method uses Sauvola's approach to estimate the background surface, it shows similar types of errors to those that appear in Sauvola's method. In spite of the complicated procedure and post-processing using shrink and swell filters, the binarization quality by Gatos' approach is barely improved. The proposed method is robust to the drastic change of illumination and reflected light. In addition, it has solved the problem of broken strokes and ghost objects, so it shows the best results.

V. CONCLUSION

This study has proposed a new technique for thresholding license plate images. The previous methods have revealed some problems in thresholding shadowed license plate images. Two major problems are broken strokes and ghost objects appearing at the boundaries of shadows. One additional problem is a thresholding error appearing at highlights due to reflected light. The proposed method has solved the broken stroke problem by adaptively adjusting the window size and location. Most of the ghost objects have also been removed by using the post-processing method based on cluster analysis. The error caused by reflected light has been easily defeated, because our method is based on Niblack's method that is robust to reflected light. The experi-

mental results show that the proposed method is superior to five well-known approaches.

REFERENCES

1. C. Arth, D. Limberger, and H. Bischof, "Real-time license plate recognition on an embedded DSP-platform," in *Proc. of CVPR* (Minneapolis, USA, June 2007), pp. 1-8.
2. P. Comelli, P. Ferragina, M. N. Granieri, and F. Stabile, "Optical recognition of motor vehicle license plates," *IEEE Trans. on Vehicular Technology* **44**, 790-799 (1995).
3. N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. on SMC* **9**, 62-66 (1979).
4. Y.-Q. Yang, J. Bai, R.-L. Tian, and N. Liu, "A vehicle license plate recognition system based on fixed color collocation," in *Proc. of the 4th ICMLC* (Guangzhou, China, Aug. 2005), pp. 5394-5397.
5. F. Yang, Z. Ma, and M. Xie, "A novel binarization approach for license plate," in *Proc. of Industrial Electronics and Applications* (Singapore, May 2006), pp. 1-4.
6. J. Bernsen, "Dynamic thresholding of grey-level images," in *Proc. of ICPR* (Paris, France, Oct. 1986), pp. 1251-1255.
7. W. Niblack, *An Introduction to Digital Image Processing* (Prentice Hall, Englewood Cliffs, NJ, USA, 1986), pp. 115-116.
8. J. Sauvola and M. Pietikainen, "Adaptive document image binarization," *Pattern Recognition* **33**, 225-236 (2000).
9. X.-Y. Yang, K.-L. Kim, and B.-K. Hwang, "An efficient binarization method for vehicle license plate character recognition," *Journal of Korea Multimedia Society* **11**, 1649-1657 (2008).
10. H.-C. Tan and H. Chen, "A novel car plate verification with adaptive binarization method," in *Proc. of the 7th ICMLC* (Kunming, China, July 2008), pp. 12-15.
11. B.-F. Wu, S.-P. Lin, and C.-C. Chiu, "Extracting characters from real vehicle license plates out-of-doors," *IET Computer Vision* **1**, 2-10 (2007).
12. I.-J. Kim, "Multi-window binarization of camera image for document recognition," in *Proc. of IWFHR-9* (Tokyo, Japan, Oct. 2004), pp. 323-327.
13. Q. Huang, W. Gao, and W. Cai, "Thresholding technique with adaptive window selection for uneven lighting image," *Pattern Recognition Letters* **26**, 801-808 (2005).
14. B. Gatos, I. Pratikakis, and S. J. Perantonis, "Adaptive degraded document image binarization," *Pattern Recognition* **39**, 317-327 (2006).
15. O. D. Trier and A. K. Jain, "Goal-directed evaluation of binarization methods," *IEEE Trans. on PAMI* **17**, 1191-1201 (1995).
16. M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic Imaging* **13**, 146-165 (2004).
17. Y. Yang and H. Yan, "An adaptive logical method for binarization of degraded document images," *Pattern Recognition* **33**, 787-807 (2000).
18. P. Stathis, E. Kavalliatou, and N. Papamarkos, "An evaluation survey of binarization algorithms on historical documents," in *Proc. of ICPR* (Florida, USA, Dec. 2008), pp. 1-4.
19. Y. J. Zhang, "A survey on evaluation methods for image segmentation," *Pattern Recognition* **29**, 1335-1346 (1996).