Best Combination of Binarization Methods for License Plate Character Segmentation

Youngwoo Yoon, Kyu-Dae Ban, Hosub Yoon, Jaeyeon Lee, and Jaehong Kim

A connected component analysis from a binary image is a popular character segmentation method but occasionally fails to segment the characters owing to image noise and uneven illumination. A multimethod binarization scheme that incorporates two or more binary images is a novel solution, but selection of binarization methods has never been analyzed before. This paper reveals the best combination of binarization methods and parameters and presents an in-depth analysis of the multimethod binarization scheme for better character segmentation. We carry out an extensive quantitative evaluation, which shows a significant improvement over conventional single-method binarization methods. Experiment results of six binarization methods and their combinations with different test images are presented.

Keywords: Character segmentation, binarization, binary image, automatic license plate recognition.

Manuscript received Aug. 13, 2012; revised Feb. 18, 2013; accepted Feb. 22, 2013.

This work was supported by the Converging Research Center Program funded by the Ministry of Education, Science and Technology, Rep. of Korea (2012K001330).

Youngwoo Yoon (phone: +82 42 860 5899, youngwoo@etri.re.kr), Kyu-Dae Ban (kdban@etri.re.kr), Hosub Yoon (yoonhs@etri.re.kr), Jaeyeon Lee (leejy@etri.re.kr), and Jaehong Kim (jhkim504@etri.re.kr) are with IT Convergence Technology Research Laboratory, ETRI, Daejeon, Rep. of Korea.

http://dx.doi.org/10.4218/etrij.13.0112.0545

I. Introduction

License plate recognition systems are used for such purposes as vehicle surveillance, parking management, and law enforcement. Problems with automatic license plate recognition systems have been widely investigated. Intensive research has resulted in excellent license plate recognition algorithms, and many successful commercial products have been applied to real situations with desirable results. However, many traditional license plate recognition algorithms assume a strictly controlled environment, such as the cameras being aligned to the lane of the road allowing the exact frontal image of the license plate to be captured. Furthermore, there have been many cases in which a switch is embedded on the road to trigger the capturing of a vehicle image at a precisely controlled position.

As automatic license plate recognition technology has been adopted in various areas of application, it has become obvious that the systems can benefit from more robust recognition algorithms that can deal with more complex capturing conditions. Dealing with various types of illumination or severe perspective distortions allows far wider areas of application for this technology. The recent proliferation of automatic license plate recognition research seems to have been in response to this application potential.

Most camera-based character recognition (CR) systems use a sequential process of target region detection, character segmentation (CS), and CR. A license plate recognition system finds license plates from natural scenery images, and the CS step is then used to find the position of the characters from the license plate image. Lastly, a CR step recognizes each isolated character. Among these three main steps, the robustness of CS is quite important in uncontrolled environments. Unfortunately, uneven illumination and blurring noise nullify many desirable traditional features for CS.

Among the three steps used in the license plate recognition procedures, this paper deals mainly with the CS step. Several CS methods were introduced in previous works, and Anagnostopoulos and others surveyed these methods [1]. According to the survey [1], a connected component analysis (CCA) and a vertical projection of the pixels are two popular CS methods. Many researchers have adopted such methods with additional mathematical morphology operators or heuristic post-processes to eliminate noise and broken character edges [2]-[4].

Most CS methods assume a binary image in which the pixels are classified into the foreground or background. Thus, the performance of a binarization method, that is, how well it converts grayscale images into binary images, is crucial since the overall accuracy of the CS process depends heavily on the quality of the binary images. However, uneven illumination, blurring, noise, and shadows make it difficult to distinguish foreground regions from background regions. Several local thresholding binarization methods that utilize different threshold values for each subregion have been proposed and show promising results [5]-[8]. However, even with binarization using a local thresholding method, it is difficult to isolate all of the characters correctly when the quality of the input images is inconsistent. To overcome this problem, the use of multiple binary images was proposed for license plate detection [9], and a subsequent study on license plate recognition used a multimethod binarization scheme for better CS accuracy [10]. That particular study [10] on CS found candidates of character regions from two or more different binary images and rejected false candidates using character arrangement and appearance information. The authors reported that multimethod binarization showed better CS accuracy compared to a single-method binarization, which uses only one binary image.

While these previous studies [9], [10] that discussed multimethod binarization used several different binarization methods, there have been no in-depth analyses or experiment results on selecting the best binarization methods and their corresponding parameters. The study in [9] introduced a framework for multimethod binarization and used 10 binary images generated through global thresholding binarization methods; however, no comparisons with different binarization methods were made.

Herein, we conduct a series of experiments to compare different binarization methods in a multimethod binarization framework and present the best combination of such methods and their parameters for CS in terms of automatic license plate recognition. Regarding CS accuracy, using a single binary image is compared with using multiple binary images. In addition, we evaluate the binarization methods regarding the different quality of images. Section II introduces the idea of multimethod binarization, as well as each binarization method used in our experiment. Section III describes a multimethod binarization scheme for character segmentation. Section IV presents the experimental design, and section V presents the experiment results and the analysis. The paper is concluded in section VI.

II. Related Works

1. Multimethod Binarization Scheme

A multimethod binarization scheme was proposed for detecting vehicle license plates from natural scene images [9]. The overall procedure of detecting the plates consists of three steps. The first step is to binarize a grayscale image into binary images with multiple global thresholds. Secondly, a region labeling algorithm finds bounding boxes of connected regions in the binary images. The bounding boxes from different binary images can overlap, so one blob of a higher compactness value among overlapped blobs is selected to reduce complexity and ambiguity. Compactness value is defined as the ratio of the foreground regions to its bounding box. Lastly, the algorithm selects four main character blobs among the candidate blobs. Every group of four horizontally adjacent blobs is tested by a binary classifier whether the blobs are character blobs or not. We assume that the groups classified as containing character blobs are license plate regions. The classifier uses a feature vector that represents the horizontal and vertical distances between adjacent blobs and the average intensity values of foreground regions and background regions. In other words, the classifier considers pixel intensities and arrangement of the blobs. The classifier is trained with a set of images that has ground truth of character regions.

2. Binarization Methods

Binarization methods can be categorized into two classes of thresholding methods: global and local. Global thresholding methods, which use a single threshold value for the entire image, do not provide satisfactory results when the image has uneven illumination or shadows. The method by Otsu is the most popular global thresholding method and chooses a threshold to minimize the intra-class variance of the foreground and background pixels [11]. Kittler and Illingworth also proposed a global thresholding method, which finds a threshold by modeling the foreground and background into two Gaussian distributions [12]. A detailed description of these

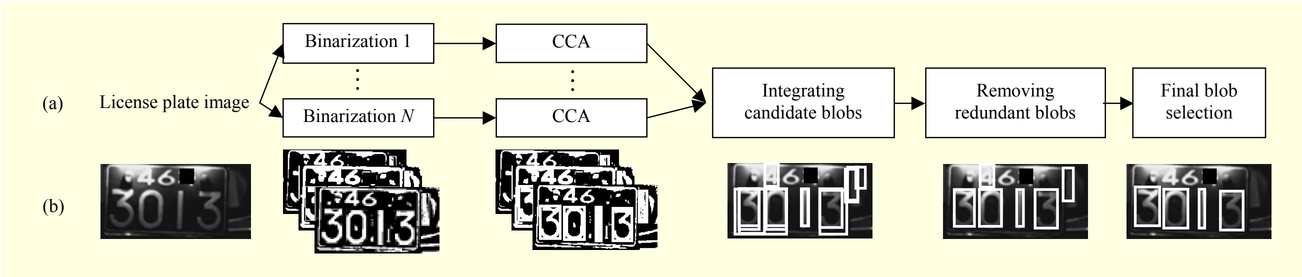

Fig. 1. (a) Flow diagram of multimethod-binarization-based character segmentation and (b) sample images of each step. Procedure inputs license plate image and finds bounding boxes of main characters. White bounding boxes in images represent segmented character blobs.

two methods can be found in the survey in [13].

Because there are many local thresholding binarization methods, we briefly introduce four popular local thresholding methods used in our experiment. Basically, the local thresholding methods calculate a threshold for each pixel by inspecting the surrounding region. In a method proposed by Niblack [6], the pixel-wise threshold, T, is computed using the mean, m, and standard deviation, s, of a surrounding region, which is a square window with a width of N:

$$T = m + k * s, \tag{1}$$

where k is a weight variable controlling the overall level of the thresholds. In this method, N and k are adjustable according to the target images. Sauvola and Pietikäinen [5] and Wolf and others [7] proposed different formulae for pixel-wise thresholds. Sauvola and Pietikäinen used the following formula:

$$T = m * [1 + k * (s / R - 1)], \tag{2}$$

where R is the dynamic range of the standard deviation. Sauvola and Pietikäinen's method was designed to amplify the contribution of the standard deviation, and it reputably provides better results than Niblack's method for a noisy and uneven background. Wolf and others proposed the following formula:

$$T = m + k * (s / R - 1) * (m - M), \tag{3}$$

where M is the minimum intensity value of the input image. Wolf and others claimed that this method solved the problems of the thinner edges and holes that occur from Sauvola and Pietikäinen's method.

Bernsen also developed a popular local thresholding method, computing the thresholds as follows:

$$T_1 = [\max(I_{sub}) + \min(I_{sub})] / 2, \tag{4}$$

where "max" and "min" indicate the maximum and minimum intensity values of the surrounding region, I_{sub} [8]. An improved version of Bernsen's method was proposed by Wen and others to segment characters in license plate images with sudden intensity changes from existing shadows [14]. This improved version (often referred to in this paper as "Wen," for short) uses a blurred image, I', which is processed using a Gaussian filter, and combines two thresholds from the original and blurred images:

$$T_2 = [\max(I'_{sub}) + \min(I'_{sub})] / 2, \tag{5}$$

$$T = \beta * [(1 - \alpha) * T_1 + \alpha * T_2], \tag{6}$$

where α is the weight from Wen and others' method after applying a Gaussian filter. When α is equal to 0, Wen and others' method is exactly the same as Bernsen's method. β controls the overall threshold levels, as does k.

III. Multimethod Binarization for Character Segmentation

In section II, we briefly reviewed the multimethod binarization scheme that was designed for detecting license plates. In our previous work [10], we improved the multimethod binarization scheme for CS using local thresholding binarization methods and CR results. Figure 1 shows the process of the CS module. The five steps are as follows: generating multiple binary images, CCA, integrating candidate blobs, removing redundant blobs, and selecting the final blobs. First, the module generates multiple binary images using different binarization methods and parameters, and we expect the multiple binary images to have successfully isolated characters at different image qualities. Lee and others [9] used 10 global thresholding methods, and four local binarization methods were used in our previous work [10]. Secondly, CCA finds the candidate blobs of characters, which are depicted as bounding boxes in Fig. 1, from each binary image. Third, the candidate blobs from multiple binary images are gathered and made into a larger set of candidate blobs. The integrated candidate set may have two or more candidate blobs for a character, as shown in Fig. 1, and it is therefore important to select the correct blobs among these many candidates. Before selecting correct blobs, we remove redundant blobs. The algorithm preserves only one blob among a set of overlapping blobs. We consider two blobs to be overlapping if the overlapping area is bigger than half of any one blob's area. To select one representative blob, we use the CR module, which outputs a single set of recognized characters and the corresponding likelihood score. The blob that has the maximum CR score is selected, and the others are removed. Blobs with low CR scores are also removed, even if there is no overlapping.

The last step is the selection of the four (or six) main characters among the candidate blobs. Blob arrangement is only cued to select main characters because pixel intensity and appearance is considered in the early steps. When we have N candidates, we test all combinations of the four (or six) blobs (total number of tests = ${}_N C_4$ or ${}_N C_6$). A test is done by a trained classifier that inputs a feature vector including the values of horizontal and vertical distances between adjacent blobs, the widths of blobs, and the heights of blobs. The output of the classifier is a binary value indicating whether the set of blobs is a correct character set or is an incorrect character set. When more than two groups are classified as containing correct character blobs, a group with a higher classification probability is selected.

CS with multiple binary images has a big advantage in handling images of different quality, but a step for selecting the final blobs among the candidates is needed. License plates have a predefined arrangement of numbers, and there are thus few errors in the blob selection stage. However, when the cues to select the blobs are insufficient, CS with multiple binary images is difficult to use. In addition, binarization methods that are complementary to one another should be selected. The use of more binary images can increase the chance that candidate blobs will include true character regions, which also increases the time complexity and possibility of selecting incorrect candidates. It is preferable to use fewer binary images when discovering all true character blobs.

IV. Experimental Design

Through experimentation, we evaluate the combinations of binary images based on other binarization methods and parameters. In these experiments, CCA-based CS is used to perform an objective evaluation of the binarization results.

1. Test Images

A total of 3,281 license plate images are collected for the experiment. These images are cropped from the natural scene images captured in three different cameras in different environments. The license plate images are transformed into 200×100 pixels. As shown in Fig. 2(a), the set of images includes not only plates in desirable condition but also some

Fig. 2. Sample images in test database, including (a) original, (b) downsampled, (c) Gaussian blurred, (d) motion blurred, and (e) shadowed.

plates with dirt and uneven illumination. Among a total of 2,044 plates, two color schemes appear: one scheme is four white numbers against a black background, and the other scheme is six black numbers against a white background.

To evaluate the methods under harsher conditions, additional experimental images generated by adding four different types of artificial noise are included. Figures 2(b) through 2(e) show sample images with noise. The first is a set of downsampled images. Low-resolution images are artificially made by downsampling the original images. The images are downsampled to 40×20 and then resized to 200×100 to make blocking artifacts. The second set includes Gaussian blur, which often occurs from a problem with the camera focus. Gaussian blur is generated by applying a circular averaging filter with a randomly selected radius of 5 to 10. Motion blur is also a common type of noise occurring in imaging systems. Shooting a fast moving object at a slow shutter speed often results in motion blur. We artificially add linear motion blur for each image in a randomly selected direction and a length of 10 to 20. Finally, images with shadows are made by blending the plate images and the black shadow images. As in real environments, all shadows lie horizontally, and their vertical positions are randomly selected within a range of the upper half of the images. In total, 16,405 images ($3{,}281 \times 5$) are used in the experiments.

Table 1. List of binarization methods and their parameters used in experiment. Three global and four local thresholding methods tested, and selected range of parameters used. Otsu and Kittler [11], [12] do not use any parameters.

Category	Binarization methods (abbreviation)	Parameters
Global thresholding methods	Fixed threshold (FixedTh)	th = [50, 65, 80, 95, 110, 125, 140, 155, 170, 185]
	Otsu's method (Otsu) [11]	-
	Kittler and Illingworth's method (Kittler) [12]	-
Local thresholding methods	Niblack's method (Niblack) [6]	N = [11, 21, 31, 41, 51, 61] k = [–0.4, –0.2, 0.0, 0.2, 0.4, 0.6, 0.8, 1.0]
	Sauvola and Pietikäien's method (Sauvola) [5]	
	Wolf and other's method (Wolf) [7]	
	Wen and others' method (Wen) [14]	N = [11, 21, 31, 41, 51, 61] α = [0.2, 0.6, 1.0] β = [0.6, 0.8, 1.0]

2. Binarization Methods

Table 1 shows the binarization methods and parameters used in these experiments. We select three global thresholding methods and four local thresholding methods based on their popularity and accuracy. FixedTh is the most basic global thresholding method, as it binarizes an input image with a predefined threshold value. Lee and others [9] used fixed thresholds for multimethod binarization, so we also test FixedTh. Otsu and Kittler have been used in many CS studies [1], [3], [13]. Niblack is a widely used local binarization method for CS [4], [15], and Sauvola is reputed to have a higher accuracy rate than Niblack [5]. Wen is a recently proposed method for CS of license plates [14]. Wolf is also a recently proposed method and reputed to be superior to Niblack and Sauvola [7].

FixedTh receives a predefined threshold value. Otsu and Kittler each automatically calculate the threshold value for a given image, and they therefore do not input any parameters. Niblack, Sauvola, and Wolf each obtain the same parameters of N and k, but Wen uses N, α, and β. We use a fixed R value of 128 for Sauvola since R does not significantly affect the resulting binary images [15]. The standard deviation of the Gaussian filter used in Wen is 3.0.

The descriptions of the binarization methods in section II assume that the character is darker than its background. However, in some license plate images, the character region is white in color, and an inverting operation is therefore added to the output binary images to obtain the same images in which each character has a white color. For the local binarization methods, we multiply the k and β values by –1 to obtain the correct binarization results for the case in which the character is brighter than the background. We assume that the type of color scheme is known and that there are therefore no CS errors caused by a faulty color estimation.

3. Evaluation Procedures

A common metric for the evaluation of binarization is misclassification errors, in which the output binary image and a manually labeled binary image are compared at the pixel level [13]. However, we are attempting to solve the CS problem, not a general binarization problem, and measuring the CS accuracy is therefore more adequate. This goal-directed evaluation method was introduced before and has been used in many studies [4], [14], [16]. In these experiments, we evaluate the binarization methods at the stage of integrating the candidate blobs, as shown in Fig. 1. The steps of removing overlapping blobs and selecting the final blobs are not applied in these experiments, to reduce the effect of steps not relevant to binarization. Similar to its use in previous studies, we use the CCA to obtain the candidate blobs of the characters from binary images.

In our evaluation of the binarization methods, we calculate the hit rates instead of the misclassification errors. Hit rates are defined as N_{included} / N_{total}, where N_{included} is the number of true bounding boxes of characters included in a set of candidate blobs, and N_{total} is the total number of characters. The number of candidate blobs should be manageable. While perfect hit rates may be reflected in a large collection of candidate blobs, having a large number of candidates might be at the expense of an increase in confusion and complexity in the subsequent selection of the final blobs. Therefore, having higher hit rates among fewer candidate blobs is preferred. We manually tag the true bounding boxes of the characters to calculate the hit rates. The four (or six) large numbers in each test image are tagged. The smaller numbers and the Korean characters on the license plates are not considered in these experiments.

Various binarization methods and parameters are also tested along with different test images in terms of the hit rates and number of candidate blobs. Different combinations of binarization methods and parameters are also tested. The number of binary images in a combination varies from two to four. The results of single-method binarization are compared with those of multimethod binarization.

V. Experiment Results and Analysis

1. Single-Method Binarization

Before evaluating the framework of multimethod binarization, we evaluate the conventional single-method

Table 2. Hit rates (%) of binarization methods for given test images. Only results with best parameters showing highest hit rates are presented. "All of the above" includes all original and noise-added test images.

Test images \ Binarization methods	FixedTh	Otsu	Kittler	Niblack	Sauvola	Wolf	Wen
Original	92.78 ($th=125$)	94.29	65.39	98.67 ($N=21, k=0.4$)	98.17 ($N=21, k=0.2$)	98.23 ($N=21, k=0.4$)	97.41 ($N=11, \alpha=1.0, \beta=0.8$)
Downsampled	75.66 ($th=125$)	68.16	6.00	92.61 ($N=11, k=0.2$)	92.29 ($N=21, k=0.2$)	92.50 ($N=21, k=0.4$)	85.46 ($N=11, \alpha=0.2, \beta=1.0$)
Gaussian blurred	34.61 ($th=170$)	25.56	0.11	60.66 ($N=31, k=0.4$)	54.98 ($N=31, k=0.2$)	52.38 ($N=31, k=0.2$)	44.40 ($N=31, \alpha=1.0, \beta=0.8$)
Motion blurred	44.17 ($th=140$)	39.87	2.01	57.59 ($N=41, k=0.4$)	55.63 ($N=41, k=0.2$)	54.40 ($N=41, k=0.4$)	50.53 ($N=31, \alpha=0.6, \beta=1.0$)
Shadowed	32.97 ($th=65$)	16.77	0.87	87.38 ($N=11, k=0.2$)	77.88 ($N=11, k=0.0$)	81.55 ($N=11, k=0.2$)	68.75 ($N=11, \alpha=1.0, \beta=0.8$)
All of the above	51.32 ($th=140$)	48.93	14.88	74.38 ($N=21, k=0.4$)	69.57 ($N=31, k=0.2$)	68.37 ($N=31, k=0.4$)	65.86 ($N=11, \alpha=0.2, \beta=1.0$)

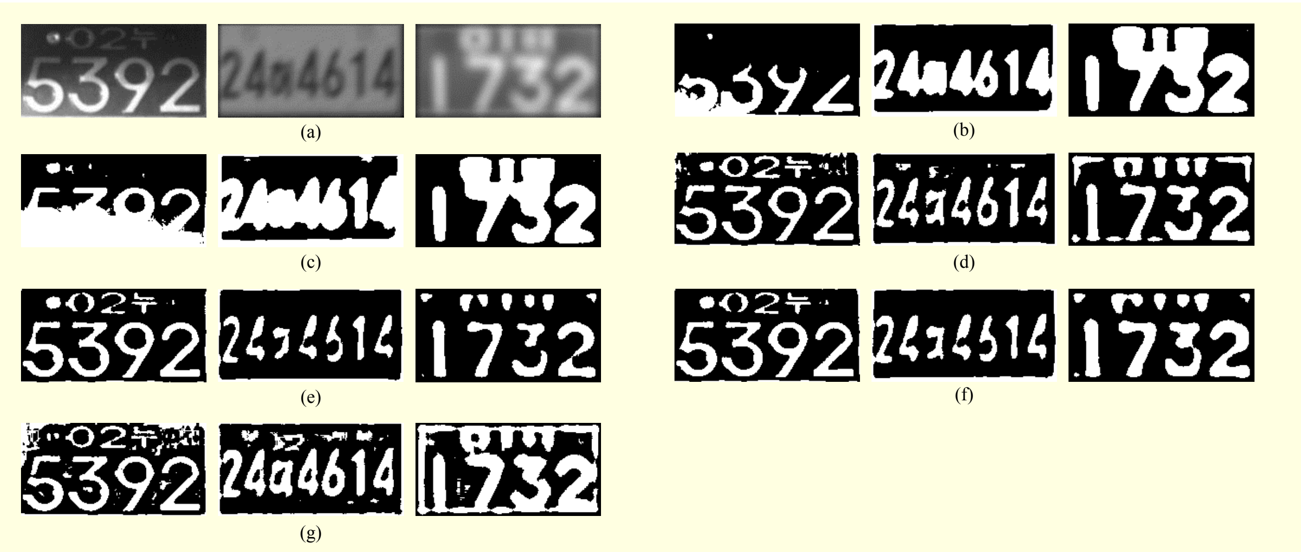

Fig. 3. Sample results of binarization methods. Results from three sets of (a) input images as well as images from (b) Otsu, (c) Kittler, (d) Niblack ($N=21$, $k=0.3$), (e) Sauvola ($N=1$, $k=1$), (f) Wolf ($N=1$, $k=1$), and (g) Wen ($N=1$, $\alpha=3$, $\beta=1$) are presented. Parameters showing best CS performance for all images in test database are used.

binarization methods. Table 2 shows the hit rates of the binarization methods with their best parameters. As reported in previous works [13], [16], the local binarization methods show better results, particularly in images with noise. In this experiment, Niblack performs best for all sets of test images. Sauvola and Wolf rank second and third, respectively, for all images. We also find that the best parameters differ along the test images. For Niblack, ($N = 11$, $k = 0.2$) is best when it processes downsampled images, but ($N = 41$, $k = 0.4$) is best with motion-blurred images.

Figure 3 shows the sample results of the binarization methods. The global thresholding methods do not work well when an input image has uneven illumination. Having one threshold value for all pixels often results in a character in a shadowed region being mislabeled as a part of the background. All of the local thresholding methods result in binary images of sufficient quality. Binary images processed by Niblack have noise on flat background regions, which does not degrade the accuracy of CCA-based CS.

In the local thresholding methods, the parameter selection is also important. Figure 4 compares the results of Niblack with different parameters of N and k, where N is the window size

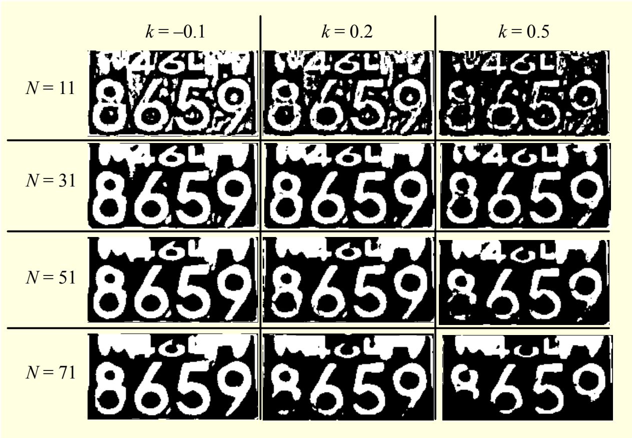

Fig. 4. Sample results of Niblack with different parameters. N = [11, 31, 51, 71] and k = [–0.1, 0.2, 0.5] are used. Input gray image also appeared in Fig. 2.

used in computing the pixel-wise thresholds. Low N values make the binarization more sensitive to local intensity changes, and the output binary images thus have rough edges. On the contrary, a high N value makes smooth binary images. The value of k increases or decreases the threshold values overall and consequently makes a larger or smaller foreground region. As shown in Fig. 4, the relatively dark character regions are labeled as parts of the background when we use a high k value.

Merged blobs and fragmented blobs are two representative errors that can occur in CCA-based CS. Merged blobs contain two or more falsely connected characters. When edges in a character are disconnected through poor binarization, the character will have several fragmented blobs. Figure 4 shows a case in which Niblack (N = 11, k = 0.5) generates fragmented blobs. With a low N value, the edges are often disconnected since weak edges in a character are labeled as parts of the background. A high k value will also increase the chance of having fragmented blobs. Meanwhile, high N and low k values generate merged blobs. Although only the results of Niblack are presented, the other local thresholding methods have the same effects based on N and k. In addition, the effect that β in Wen has is similar to the effect that k has.

As we discussed with Table 2, a single-binarization method with a fixed parameter does not give the best results for all sets of different test images. Therefore, using two or more binarization methods has the possibility to provide better CS results when each binarization isolates the characters at different image qualities. We also find that the different binarization methods and parameters result in various binary images, as shown in Figs. 3 and 4.

2. Combinations of Binary Images

Table 3 shows the hit rates of the different binarization methods. Combinations of two to four binary images are evaluated. For all sets of test images, multimethod binarization shows higher hit rates than single-method binarization. Single-method binarization shows a hit rate of 74.38% in a test of all images, while the use of four binary images generates a hit rate of 88.49%. About 20 candidate blobs are generated when we use four binary images. The hit-rate increments for the different test sets are all different, but the test set for Gaussian blur has the biggest hit-rate increment of 60.66% to 81.50%.

In the experiments, Niblack is most frequently selected as the best for different combinations. Sauvola and Wolf are also selected in certain cases, but Wen and the global thresholding method are not selected. Different parameters are selected for multimethod binarization rather than different binarization methods themselves. When four binary images are used for all test images, Niblack binarization is applied four times with four different values of N and k. The best parameters selected for single-method binarization and multimethod binarization differ. Niblack (N = 21, k = 0.4) is selected for single-method binarization, but different parameters of (N = 11, k = 0.2) and (N = 41, k = 0.4) are selected for the combination of two binary images.

When we test for all images, the best hit rate for binarization using one, two, three, and four binary images is about 74%, 84%, 87%, and 88%, respectively. It is evident that the hit rate increases as we use more binary images, but its rate of increase decreases the more difficult the samples of the remaining failure cases are. The use of more binary images has a tradeoff between a higher hit rate and a higher number of candidates, which increases the complexity and confusion in the subsequent CS processes. Therefore, a system developer should choose a proper number of binary images according to their target images and applications.

We also analyze the binarization methods in terms of their complementarity. In the experiments, the hit rates are expressed as the coverage of correct CS. Multimethod binarization is designed to extend the coverage by combining two or more complementary binarization methods. However, maximizing the coverage is not a straightforward process. When each binarization method has a wide coverage and the combination of methods works in a complementary fashion, we can maximize the coverage and obtain higher hit rates. Working in a complementary fashion results in a low redundancy between methods, where redundancy R is defined as $R = N_{\text{correctByOne}} / N_{\text{correct}}$, in which $N_{\text{correctByOne}}$ is the number of characters correctly discovered by only one method, and N_{correct} is the total number of correctly discovered characters. When a character is discovered by two or more methods, we assume that the character is redundantly discovered.

Table 4 shows some combinations along with their hit rates

Table 3. Combinations of different binarization method: (a) best combination of two to four binarization methods, (b) hit rate (%), and (c) number of candidate blobs.

Test images \ # of combinations	2	3	4
Original	(a) Niblack (N = 11, k = 0.4) Wolf (N = 31, k = 0.2) (b) 99.67 / (c) 12.16	Niblack (N = 11, k = 0.4) Niblack (N = 31, k = 0.0) Niblack (N = 31, k = 0.6) 99.79 / 18.13	Niblack (N = 31, k = 0.0) Niblack (N = 31, k = 0.4) Niblack (N = 31, k = 0.8) Wolf (N = 11, k = 0.2) 99.87 / 23.69
Downsampled	Wolf (N = 11, k = 0.2) Wolf (N = 21, k = 0.2) 96.39 / 11.06	Niblack (N = 11, k = 0.2) Sauvola (N = 41, k = 0.2) Wolf (N = 11, k = 0.2) 97.62 / 17.29	Niblack (N = 11, k = 0.0) Niblack (N = 61, k = 0.6) Sauvola (N = 21, k = 0.2) Wolf (N = 11, k = 0.2) 98.19 / 22.99
Gaussian blurred	Niblack (N = 11, k = 0.0) Niblack (N = 41, k = 0.6) 73.01 / 6.17	Niblack (N = 11, k = 0.0) Niblack (N = 11, k = 0.2) Niblack (N = 41, k = 0.6) 78.91 / 10.21	Niblack (N = 11, k = 0.0) Niblack (N = 11, k = 0.2) Niblack (N = 51, k = 0.6) Sauvola (N = 31, k = 0.2) 81.50 / 14.05
Motion blurred	Niblack (N = 31, k = 0.2) Niblack (N = 51, k = 0.6) 63.95 / 8.33	Niblack (N = 11, k = 0.2) Niblack (N = 31, k = 0.2) Niblack (N = 51, k = 0.6) 67.67 / 13.25	Niblack (N = 11, k = 0.2) Niblack (N = 31, k = 0.2) Niblack (N = 61, k = 0.6) Wolf (N = 41, k = 0.8) 69.82 / 16.86
Shadowed	Niblack (N = 11, k = 0.0) Niblack (N = 21, k = 0.6) 95.69 / 11.20	Niblack (N = 11, k = 0.0) Niblack (N = 11, k = 0.2) Niblack (N = 21, k = 0.6) 96.69 / 17.39	Niblack (N = 11, k = 0.0) Niblack (N = 11, k = 0.4) Niblack (N = 21, k = 0.6) Wolf (N = 21, k = 0.2) 97.08 / 21.96
All of the above	Niblack (N = 11, k = 0.2) Niblack (N = 41, k = 0.4) 83.65 / 10.42	Niblack (N = 11, k = 0.0) Niblack (N = 11, k = 0.4) Niblack (N = 41, k = 0.4) 86.72 / 15.22	Niblack (N = 11, k = 0.0) Niblack (N = 11, k = 0.2) Niblack (N = 31, k = 0.6) Niblack (N = 41, k = 0.4) 88.49 / 20.33

Table 4. Hit rates (%) and redundancy (%) of multiple binarization attempts. Tests conducted using all test images.

Combinations	Hit rate	Redundancy
Niblack (N = 21, k = 0.4) Niblack (N = 31, k = 0.4)	79.53	86.94
Niblack (N = 11, k = 0.2) Niblack (N = 41, k = 0.4)	83.65	70.48
Niblack (N = 11, k = –0.2) Niblack (N = 41, k = 0.8)	74.64	40.55
Niblack (N = 11, k = –0.4) Niblack (N = 51, k = 1.0)	57.69	4.68

and redundancies. The best combination of Niblack, (N = 11, k = 0.2) and (N = 41, k = 0.4), which has the highest hit rate, shows a redundancy of 70.48%. The (N = 21, k = 0.4) and (N = 31, k = 0.4) combination of Niblack has a higher redundancy of 86.94% since the parameters are more similar than with the best combination. The (N = 11, k = –0.2) and (N = 41, k = 0.8) combination of Niblack shows a far smaller redundancy, but the hit rate is lower than with the best combination.

Combinations of two similar binarization methods and two distinct binarization methods show lower hit rates. Using similar binarization methods generates high redundancies, and they thus fail to extend the coverage from the use of a single-method binarization. On the contrary, using two distinct binarization methods generates low redundancies, but they have a low coverage since each binarization method has a very narrow coverage. Therefore, in selecting the binarization methods, the users must consider both the coverage of each method and the complementarity of the combination.

The researchers in [9] proposed a binarization method that uses 10 different binary images with fixed thresholds. Similarly, we also test the use of 10 binary images with FixedTh, as shown in Table 1. Our experiment results in a 52.79% hit rate, 26.8 candidate blobs, and 84.42% redundancy. This hit rate is

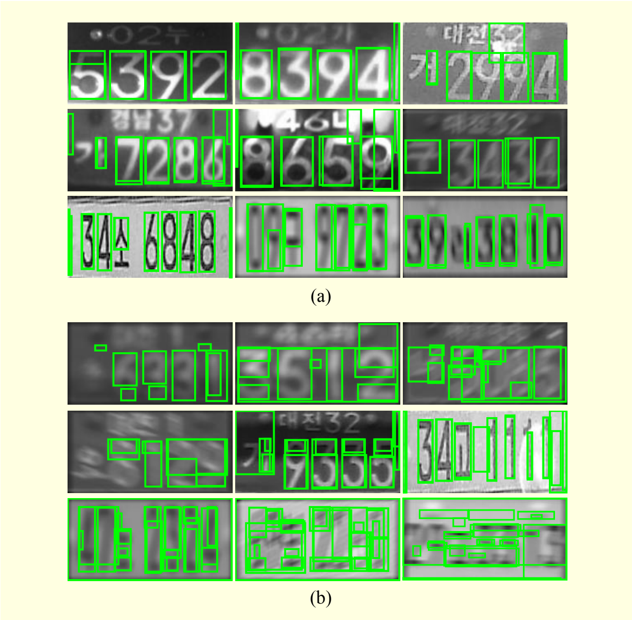

Fig. 5. (a) Correctly and (b) incorrectly segmented license plate images. Best combination of two binary images is used, and rectangular bounding boxes of candidate blobs are drawn on images.

far poorer than that which results from using two local thresholding methods. Using 10 binary images may be worse than using two binary images when the binarization methods themselves are poor and their combination with other binarization methods is not complementary. We confirm that the binarization methods and parameters should be carefully selected even when using multiple binary images.

Figure 5 shows sample results of finding candidate blobs from multimethod binarization. Correctly segmented images in which the set of candidate blobs contains all true character regions are presented. To ensure a clear view in Fig. 5(a), the extremely large and small blobs are not marked. Failed cases are also presented along with their segmentation results. Severely blurry and noisy images often generate fragmented blobs. Merged blobs also appear when blurring the artifacts connecting two characters. The average processing time of the CS and CR for a given license plate image is about 80 ms on a 3.3-GHz Pentium processor.

VI. Conclusion

A framework of multimethod binarization is a powerful approach for noisy input images; however, the selection of binarization methods and parameters in the framework has not been previously analyzed in depth. In this paper, we presented the best combination of binarization methods in a multimethod binarization framework for automatic license plate recognition. The combinations of four different selected binarization methods showed superior CS results to single-method binarization and over a combination used in a previous work [9]. A hit rate of 88.49% was achieved, while the hit rate of the best single and combined binarization in [9] was 74.38% and 52.79%, respectively. To find the best combinations, we first compared six popular binarization methods and their corresponding parameters. The combinations of the binarization methods were then evaluated. Niblack's method worked well, not only for single-method binarization but also for multimethod binarization, and its input parameters heavily affected the CS accuracy. We reported changes in the hit rates and number of candidate blobs along a number of binary images. The complementarity between binarization methods was also measured.

This study focused on CS for automatic license plate recognition, but a framework of multimethod binarization can also be used in other applications. The experiment results comparing the binarization methods and parameters with different test images are promising guidelines for selecting proper binarization methods.

References

[1] C.E. Anagnostopoulos et al., "License Plate Recognition from Still Images and Video Sequences: A Survey," *IEEE Trans. Intell. Transp. Syst*., vol. 9, no. 3, 2008, pp. 377-391.

[2] S. Nomura et al., "A Novel Adaptive Morphological Approach for Degraded Character Image Segmentation," *Pattern Recog*., vol. 38, no. 11, 2005, pp.1961 -1975.

[3] H.T. Lue et al., "A Novel Character Segmentation Method for Text Images Captured by Cameras," *ETRI J*., vol. 32, no. 5, Oct. 2010, pp. 729-739.

[4] Y. Yoon et al., "Blob Extraction Based Character Segmentation Method for Automatic License Plate Recognition System," *Proc. IEEE Int. Conf. Syst., Man, Cybern*., 2011, pp. 2192-2196.

[5] J. Sauvola and M. Pietikäinen, "Adaptive Document Image Binarization," *Pattern Recog*., vol. 33, 2000, pp. 225-236.

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

[7] C. Wolf, J.M. Jolion, and F. Chassaing, "Text Localization, Enhancement and Binarization in Multimedia Documents," *Proc. ICPR*, vol. 2, 2002, pp. 1037-1040.

[8] J. Bernsen, "Dynamic Thresholding of Gray-Level Images," *Proc. ICPR*, 1986, pp. 1251-1255.

[9] D. Lee and J. Choi, "Precise Detection of Car License Plates by Locating Main Characters," *J. Optical Soc. Korea*, vol. 14, no. 4, 2010, pp. 376-382.

[10] Y. Yoon et al., "Blob Detection and Filtering for Character Segmentation of License Plates," *Proc. Int. Works. MMSP*, 2012,

pp. 349-353.
[11] N. Otsu, "A Threshold Selection Method from Gray Level Histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 1, 1979, pp. 62-66.
[12] J. Kittler and J. Illingworth, "Minimum Error Thresholding," *Pattern Recog.*, vol. 19, no. 1, Jan./Feb. 1986, pp. 41-47.
[13] M. Sezgin and B. Sankur, "Survey over Image Thresholding Techniques and Quantitative Performance Evaluation," *J. Electron. Imaging*, vol. 13, no. 1, 2004, pp. 146-168.
[14] Y. Wen et al., "An Algorithm for License Plate Recognition Applied to Intelligent Transportation System," *IEEE Trans. Intell. Transportation Syst.*, vol. 12, no. 3, 2011, pp. 830-845.
[15] J. He et al., "A Comparison of Binarization Methods for Historical Archive Documents," *ICDAR*, 2005, pp. 538-542.
[16] O.D. Trier and A.K. Jain, "Goal-directed Evaluation of Binarization Methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 12, 1995, pp. 1191-1201.

Youngwoo Yoon received his BS and MS in computer science with honors from the Information and Communications University (merged with the Korea Advanced Institute of Science and Technology [KAIST] in 2009), Daejeon, Rep. of Korea, in 2006 and 2008, respectively. He is currently a researcher at ETRI, Daejeon, Rep. of Korea. His research interests include computer vision, HRI, and HCI.

Kyu-Dae Ban received his PhD from the University of Science and Technology (UST), Daejeon, Rep. of Korea, in 2011. He has been a researcher at ETRI, Daejeon, Rep. of Korea, since 2011. His research interests include image processing and pattern recognition.

Hosub Yoon received his BS and MS in computer science from Soongsil University, Seoul, Rep. of Korea, in 1989 and 1991, respectively. He received his PhD in image processing from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Rep. of Korea, in 2003. He joined KIST/SERI, Daejeon, Rep. of Korea, in 1991 and transferred to ETRI, Daejeon, Rep. of Korea, in 1999. His major research interests include HRI, image processing, audio processing, and pattern recognition.

Jaeyeon Lee received his PhD from Tokai University, Tokyo, Japan, in 1996. He has been a research scientist at ETRI, Daejeon, Rep. of Korea, since 1986. His research interests include robotics, pattern recognition, computer vision, and biometric security.

Jaehong Kim received his PhD from Kyungpook National University, Daegu, Rep. of Korea, in 2006. He has been a research scientist at ETRI, Daejeon, Rep. of Korea, since 2001. His research interests include socially assistive robotics for elder care, human-robot interaction, and gesture/activity recognition.