Uniform-Invoice Number Extraction

Cheng-Hua Chien and Daw-Tung Lin*

Department of Computer Science and Information Engineering

National Taipei University, Taiwan

Abstract Invoices are widely used in our daily life. Taiwan government uniform-invoice not only proves as the receipt of purchasing, but also is utilized as a lottery for bonus. However, checking the award-winning numbers is generally engaged manually and is a laborious job yet time consuming. This work presents an automatic invoice number extraction system with high accuracy. The proposed extraction system consists of two parts: invoice number detection and candidate invoice number segmentation. The developed system has been evaluated on 191 invoice images and achieved superior correct extraction performance up to 98.42%, while the false positive rate is as low as 0.094 per sample.

1 Introduction

Taiwan government uniform-invoice has been closely related to people in the daily life. The uniform-invoice not only proves as the receipt of purchasing, but also utilized as a lottery for bonus. The Ministry of Finance releases the winning numbers of uniform-invoice prizes every tow months. Checking the award-winning numbers is an interesting activity. Generally, people match the invoice numbers manually. However, when the amount of invoices is huge up to hundreds or more, it could be a laborious job and time consuming. Worst case may occur with fault reading or mis-matching and thus will loose the prize up to two million Taiwan dollars. By adopting the digital image processing technology would improve the drawback of manually checking. We can capture the invoice image by a small camera and extract the invoice number immediately. Figure 1 shows a sample invoice image. The advantages are that it can speed up the checking task, and reduce mistakes. However, the complex printed materials on the invoice may reduce the accuracy of matching. Different invoice background mark color, invoice number, store stamp, company logo, and information of the merchant items, may affect the detection. Hence, the design of number extraction algorithms is critical for accurate number identification.

The main tasks of the invoice number extraction system are invoice number detection and feature extraction. Many computer-vision-based text detection methods have been proposed. *Zhong et al.* propose two simple methods to locate text in complex images [1]. The first approach is mainly based on finding connected monochrome color regions of a certain size, while the second locates text based on its specific spatial variance. *Gao et al*. use horizontal and vertical projections of edges to localize text strings [2]; however, it can only handle captions and cannot deal with complex text layouts. A further method proposed by *Ohya et al.* is based on adaptive thresholding to segment the image in regions [3]. A couple of methods use mathematical morphology in the detection step.

* Corresponding author.

Hori dilates Sobel edges into text regions [4]. The emphasis of his work is on the binarization and removal of complex backgrounds. Hasan and Karam detect edges utilizing morphological operator [5]. Most other techniques can be classified as edge detection [4,6-8], color feature segmentation [9-11], or texture based detection [12,13,19]. For feature extraction, algorithms such as Connected Component Analysis [14,15], and Run-Length smearing [16,20] have been adopted. Similar techniques can be referred to vehicle license plate locating. For example, *Bai et al*. applied Sobel detection, thresholding, nonlinearing filter and morphology [17]; and *Zhu et al*. utilized RGB color features and morphology to resist the complex background and lighting change [18].

This work presents an invoice number extraction system with high correct segmentation accuracy. The proposed extraction system consists of two stages: invoice number detection and candidate invoice number segmentation. Figure 2 presents the flow chart of the proposed extraction system. Sections 2 and 3 illustrate these procedures, respectively. Section 4 reports the experimental results. Finally, Section 5 draws the conclusions.

2 Invoice Number Detection

A typical uniform-invoice contains many printed characters and symbols such as invoice numbers, merchant logo, merchandise item list and detail information, and invoice stamp, which are regarded as background. Furthermore, the background mark under the numbers is varied with four different colors: red, green, blue and brown depends on the month the invoices issued. The first stage tends to distinguish the invoice numbers from background. The choice of color is important to accomplish the goal. Various methods have been conducted for grayscale image conversion and will be discussed in detail in this section.

(a) (b)

Fig. 1 The invoice image sample

2.1 Grayscale Conversion based on Red Primary Color

The choice of red color is based on its physics optics. The wavelength of red light is longer than the other colors, so it is most impossible for red light to be refracted in a non-vacuum situation. Namely, red color possesses more complete information of an image than other colors do. Thus, we convert the image into grayscale based on red primary color. Afterwards, the invoice number is detected simply by thresholding.

2.1.1 Adaptive Thresholding

Adaptive thresholding is a well-known algorithm which is a histogram approach assumes that there is some average values for the background and object pixels, but that the actual pixel values have some variation around these average values. The procedure is shown in below.

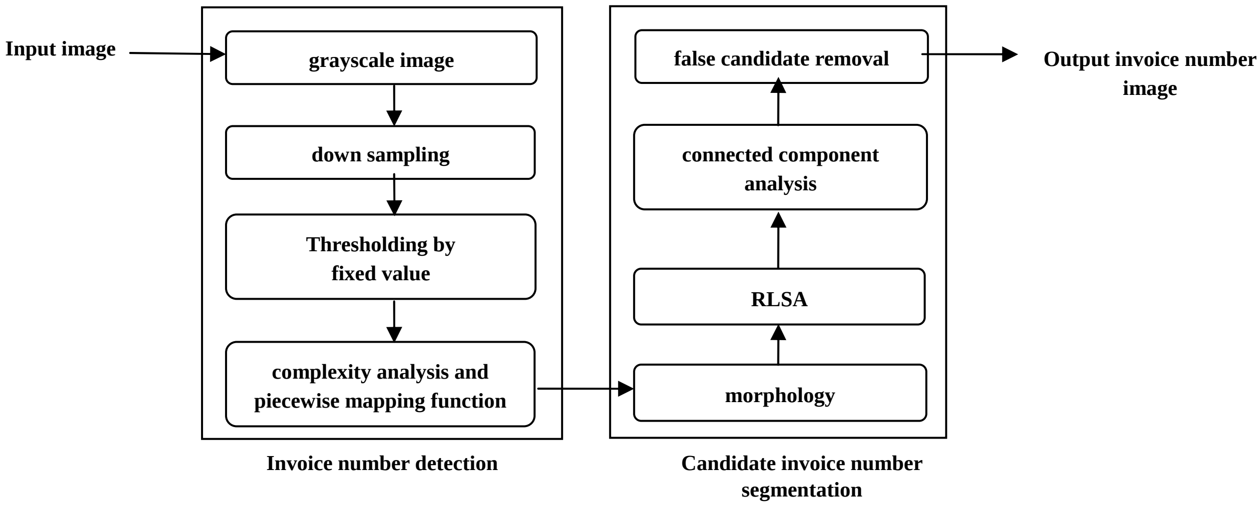


Fig. 2 Flow chart of the proposed extraction system

Step 1: Let G_O ={ $I(x,y): I(x,y)$ >T}, where G_O represents the object pixels.
Let G_B ={ $I(x,y): I(x,y) \leq$ T}, where G_B stands for the background pixels.

Step 2: Let m_O be the average value of G_O, m_B be the average value of G_B.

Step 3: Set T'= ($m_O + m_B$)/2.

Then replaces T with T' and repeats these steps until T' is similar to T.

However, this threshold is not suitable for the situation when the graylevel of invoice number pixels is closed to that of background. For example, if an invoice image has light background, adaptive thresholding will result in under-estimate of the threshold value because it neglects the minority which is complex enough and affects the detection of invoice number. Figure 3(b) shows the result using adaptive thresholding on Fig. 3(a), while Fig. 3(c) is the result with fixed value thresholding. Apparently, Fig. 3(c) obtained better binarization for invoice numbers. Figure 3(b) has a loose threshold and the value is pulled down by the clear background in the original image with adaptive thresholding method.

(a) (b) (c)

Fig. 3 (a) Sample invoice image. (b) The resultant image of adaptive thresholding. (c) The resultant image with fixed value thresholding.

2.1.2 Fixed Thresholding

Based on extensive experiments, we choose a fixed thresholding value as following (Eq. (1)):

$$B(x,y) = \begin{cases} 1, & G(x,y) \leq Th \\ 0, & G(x,y) > Th \end{cases} \tag{1}$$

where *G(x,y)* is the grayscale image as described in Section 2.1, and *Th* represents the threshold 80. However, it may not be the best thresholding value. Because sometimes invoice number may become too light in two situations: (1) different printers and (2) color fading due to folding or staining and could leads to inaccurate detection and affects the extraction performance. Hence, instead of removing more noises, the first stage tends to retain the invoice number information as much as possible by using higher threshold. Unfortunately, this higher threshold value is yet not good enough for the images with complex background of extra stamp as shown in Fig. 4.

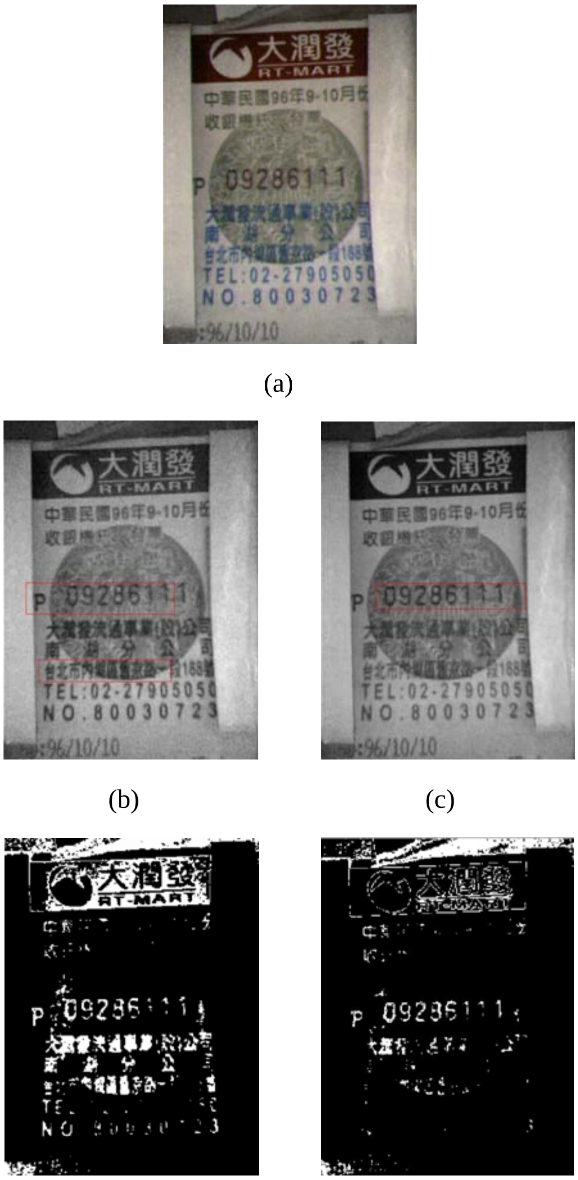
(a) (b)

Fig. 4 Example invoice images with complex background of extra stamps.

2.2 Grayscale Conversion with Weighted RGB

To improve the thresholding result and number detection, weighted RBG method is chosen to get the grayscale image. By setting equal weight on RGB components, the grayscale image is close to shading value, or says, the value of HSV. In this study, the weights on RGB are set to 0.35, 0.3, and 0.35 respectively, to emphasize the invoice number characters. And then, we obtain a binary image with the fixed thresholding as described in Section 2.1.2. The improvement is shown in Fig. 6(c). Compare to Fig. 6(b), Fig. 6(c) presents more clear invoice number, and is extracted corrected as plotted in red lines with no false positive detection. Figure 6(e) shows the binary image of Fig. 6(c) using fixed thresholding which suppresses the noise, repairs the invoice number and thus improves the detection.

Although the above mentioned method improves the thresholding results, we still need to solve the challenging issue of complex background circle mark. So far, we are not yet able to separate the invoice number from the complex circle mark background or extra store stamps. Next, contrast strength technique is adopted to enlarge the contrast between object and background.

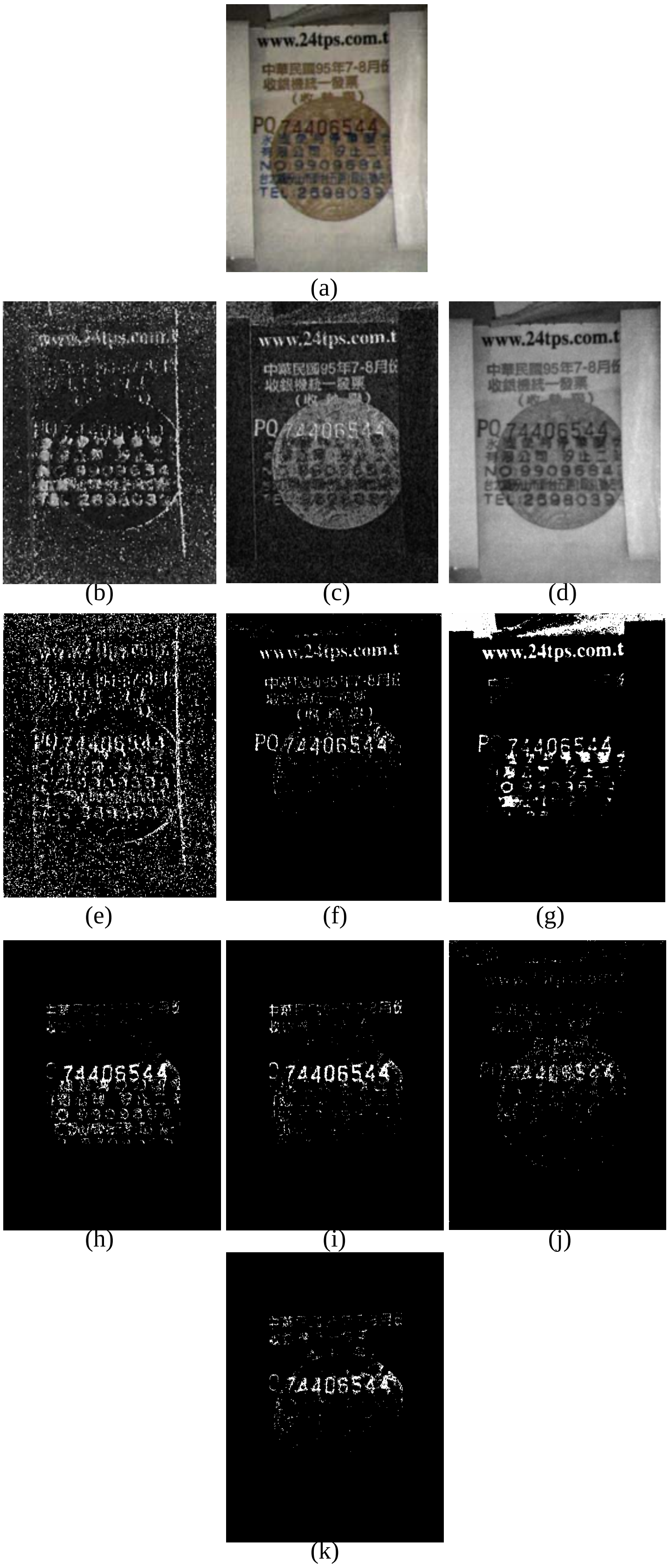


(a)

(b) (c)

(d) (e)

Fig. 5 (a) Example invoice image. (b) Graylevel image converted from red color component. (c) Graylevel image converted using weighted RGB method. (d) Fixed thresholding of (b). (e) Fixed thresholding of (c).



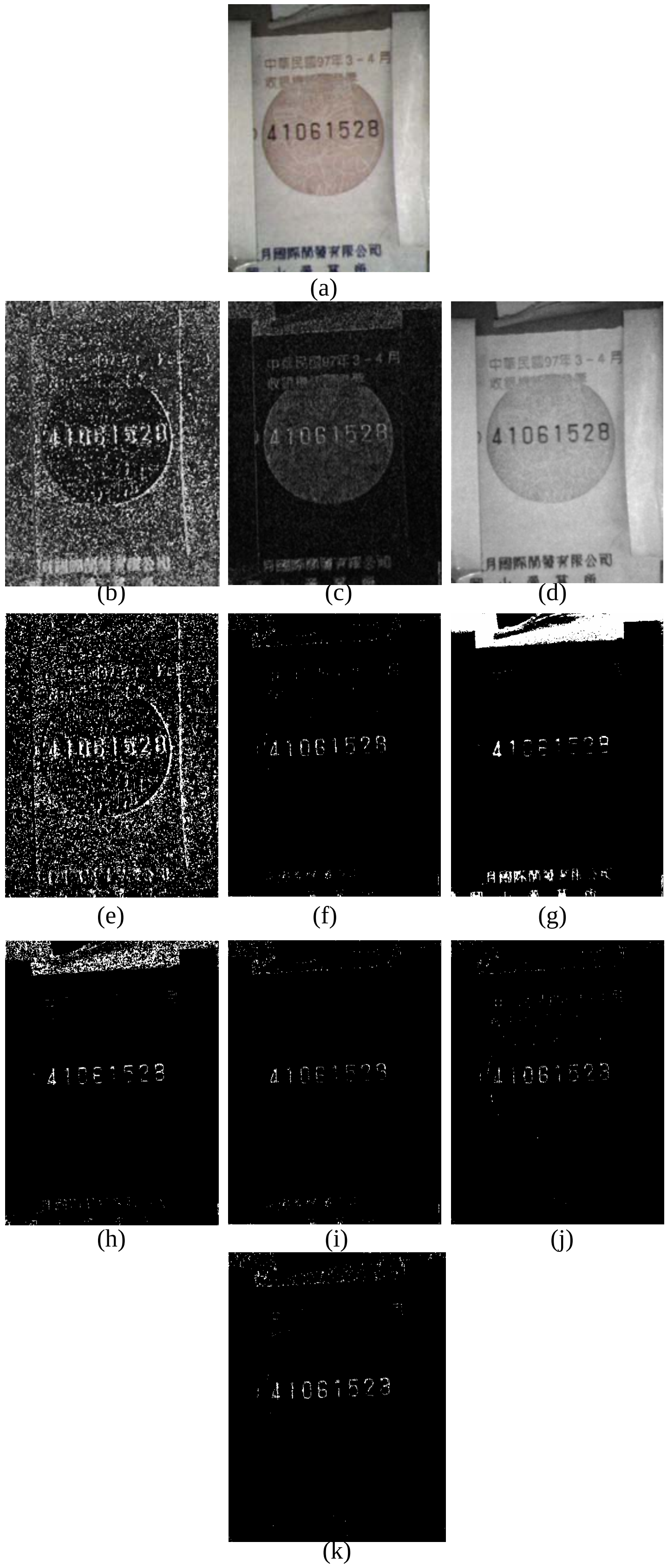


Fig. 6 (a) Example invoice image with complex background. (b) H component. (c) S component. (d) V component. (e)~(k) are the results of thresholding indicated in Table 1.

Fig. 7 (a) Example invoice image with light invoice number. (b) H component. (c) S component. (d) V component. (e)~(k) are the results of thresholding indicated in Table 2.

2.3 HSV color model

H(hue), S(saturation), V(value) color model is a related representation of points in RGB color model that attempts to describe perceptual color relationships more accurately than RGB. This study utilizes the HSV color model representation, analyzes and compares between the image with complex background and the image with light invoice number by varying the ranges of HSV (0~255). Table 1 and Table2 indicate the conditions of various HSV values for complex background case and light invoice number case, respectively. Figures 6 and 7 display the corresponding graylevel conversions. Figure 6(a) is an example invoice image with complex background. Figures 6(e)~6(k) are the results of thresholding using the constraints indicated in Table 1. For instance, Fig. 6(e) plots the thresholding result for pixels whose H component is greater than 200 or smaller than 10. Figure 6(f) displays the thresholding result for pixels whose S component is greater than 150.

Table 1 Various thresholding constrains for image with complex background.

	(e)	(f)	(g)	(h)	(i)	(j)	(k)
H	>200 ∨ <10			>200 ∨ <10		>200 ∨ <20	>200 ∨ <30
S		>150			>110	>110	>110
V			<60	<80	<80		<90

Table 2 Various thresholding constrains for image with complex background.

	(e)	(f)	(g)	(h)	(i)	(j)	(k)
H	>180			>200 ∨ <30		>200 ∨ <30	>200 ∨ <30
S		>110			>110	>100	>80
V			<80	<90	<90		<110

As we can observe from these results, to obtain better invoice number extraction, more restrict constrains are needed to remove noise with dark background. On the contrary, for the light invoice number, hard constraints will suppress the signal of invoice number and may result in fault detection. There is always a trade off in choosing the range settings of HSV values. Therefore, HSV model is apparently not an appropriate solution.

2.4 Grayscale Conversion with Blue Primary Color and Piecewise Mapping Function

Based on the preliminary study, we learn that it is not sufficient to convert from color image into grayscale image by only choosing a single color conversion method for various complex situations. It needs additional process after thresholding to reinforce the detection. The proposed process will be discussed later.

2.4.1 Grayscale Conversion Comparison

To investigate the effect of grayscale image conversion from different primary color, we have done extensive experiments on 20 invoices. Figure 10 gives five examples with different complexity and background colors: red, green, brown and blue in Fig.’s 8(a) to Fig. 8(e). Each sample image is converted to graylevel image based on the primary color. Figures 8(a)-R, 8(a)-G, and 8(a)-B show the result of grayscale image conversion of Fig. 8(a) based on R, G, and B primary color, respectively. Analogically, Fig.’s 8(e)-R, 8(e)-G, and 8(e)-B display the result of grayscale image conversion of Fig. 8(e) based on R ,G, and B primary color, respectively. According to the comparison, the contrast between invoice number and background is obvious when using B primary color for

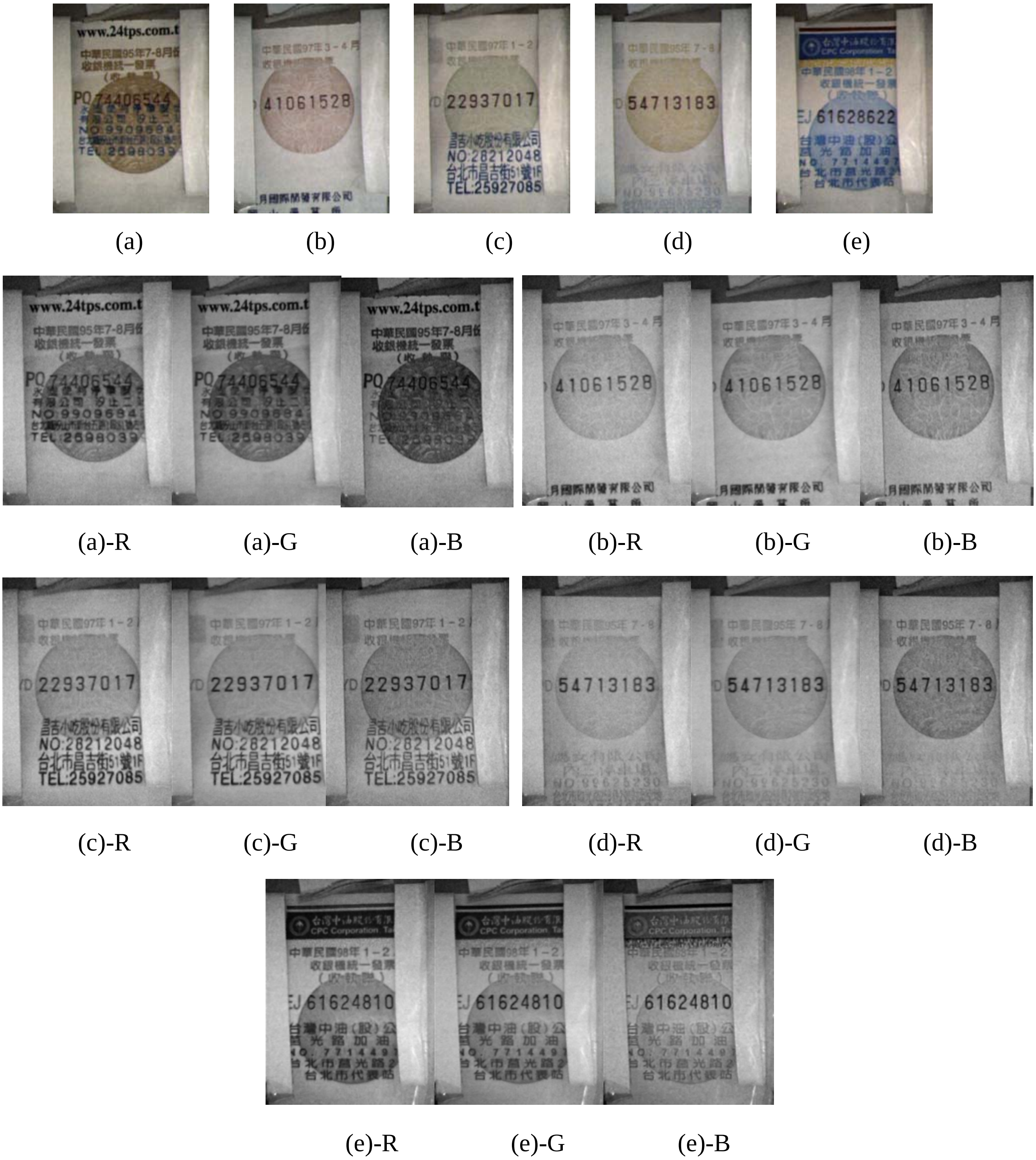


Fig. 8 Comparison of grayscale image conversion based on R, G, B, primary color for five examples. (a)-R, (a)-G, and (a)-B show the result of grayscale image conversion of (a) based on R, G, and B primary color, respectively. Analogically, (e)-R, (e)-G, and (e)-B display the result of grayscale image conversion of (e) based on R ,G, and B primary color, respectively.

grayscale image. More importantly, the store stamp becomes invisible because it is usually printed in blue ink. In summary, grayscale conversion with blue primary color is a prospective method and assists the separation between invoice number and background.

2.4.2 Linear Piecewise Stretching

This section describes the process of linear piecewise stretching and enhances the contrast of invoice number and the other printed items. When an image is converted to binary image by the method illustrated in section 2.4.1, we need to analyze and count the area of white pixels to decide if the background is still too complicated for segmentation. If the count is larger than 1000, the grayscale image would need to further stretch the contrast. We adopt the linear piecewise mapping function to enhance the contrast, i.e., enlarge the difference between the grayscale interval we are interested in. Figure 9(a) plots the linear piecewise mapping and the equation is derived as follows (Eq. (2)).

$$G1(x,y) = \frac{G(x,y)-a}{b-a}(d-c)+c \qquad (2)$$

where *G(x,y)* is the grayscale image obtained from section 2.4.1, and the four parameters a, b, c, and d are set to 10, 144, 0, and 255, respectively. This process is executed whenever the count is larger than the pre-defined threshold. Figure 9(c) presents the resultant image after contrast stretching using linear piecewise mapping function. Finding an accurate threshold value is a challenging issue due to complicate appearance and unexpected condition of images. An inappropriate choice of threshold value may cause fault detection. Usually, loose constrain will be adopted to prevent excess rejection of targets at the first place. The linear piecewise mapping function plays an important role in adjusting the difference of object pixels and the rest of signals, and thus eases the segmentation task.

(a) (b) (c)

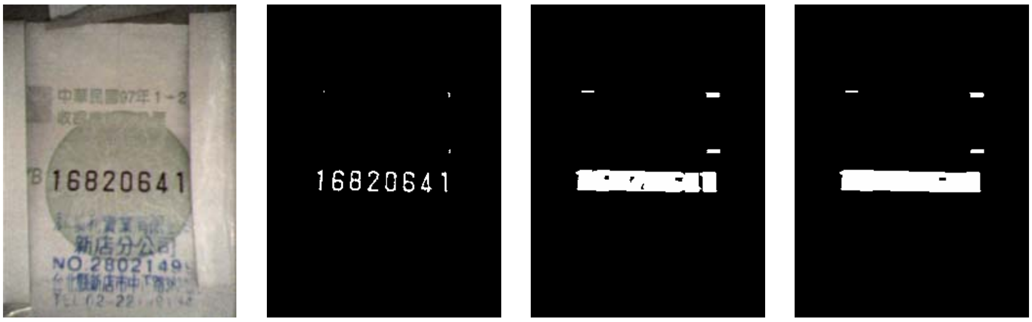
Fig. 9 (a) Linear piecewise mapping function. (b) Original grayscale image. (c) The resultant image after contrast stretching using linear piecewise mapping function.

3 Candidate Invoice Number Segmentation

After the detection and noises removal, the image remains invoice number and some fragmentary noises. Next, we need to extract the complete invoice number region involving further connecting the gaps between the numbers, and removing the invalid candidates. First, the morphology operation and run length smearing algorithm (RLSA) are utilized to fill the gaps between the numbers. Then the connected component labeling and invalid candidate justification are engaged to extract numbers and remove noises.

3.1 Gap Filling

To fill the gaps between numbers, mathematical morphology operation is utilized. The dilation operation is a powerful tool to repair the object and expend the region. Since the number region expands horizontally, a 7×1 circular structure



(a) (b) (c) (d)

Fig. 10 (a) Example invoice image. (b) The result of thresholding. (c) The result of dilation. (d) The result of RLSA.

(a) (b) (c) (d)

Fig. 11 (a) An example invoice with fragile invoice number. (b) The result of thresholding. (c) Another example of invoice image with fragile invoice number. (d) The result of thresholding.

element is applied twice to expend the region horizontally. Figure 10 depicts the procedure of invoice number candidate region segmentation. Figure 10(c) shows the result of dilation. However, a disconnection occurred because the gap is still too big (e.g. the gap beside the number 1). Hence, the RLSA is utilized to repair it. RLSA is often used to connect object pixels which are separated by less than a threshold number of background pixels to generate a smeared image of the document. In this work, RLSA is applied with parameters 6 and 1 for the threshold of horizontal and vertical smearing, respectively. Figure 10(d) presents the result of RLSA for smearing the gap between the last two numbers. As expected, the proposed method successfully extracts candidate invoice number regions.

3.2 Connected Component Analysis

After the gap filling process, it is necessary to extract the connected regions and trace it back to the original image to get the intact region of invoice number. This step determines the connected area of the remaining objects by connected component analysis and labeling. According to the connected component analysis algorithm, each connected area is assigned a label, and the locations of begin and end are stored for comparison with the original image.

3.3 False Candidate Removal

The real invoice number needs to be distinguished from cluttered objects once the candidate region has been coarsely extracted. Ideal case as shown in Fig. 10(d), only one of the candidate regions is the real invoice number. Identifying real invoice number effectively is important to avoid false positives. The proposed classification method inspects the size of area and the aspect ratio of connected components. Only the component with area size between 6000 and 27000, and the aspect ratio between 3 and 9, is identified as a real invoice number region. Since the object has certain specification, this method is sufficient to get rid of irregular noise component.

4 Experimental Results

Table 3 Evaluation results.

Correct Extraction	188	98.42% (188/191)
False Positive	18	0.094 (18/191)

To evaluate the performance of the proposed extraction method, we collected 191 invoice images of resolution 640×480 for testing. These images include different combinations of background colors, store stamps, various angles and locations. Table 3 presents the evaluation results. The proposed system achieves superior correct extraction performance up to 98.42%, while the false positive rate is as low as 0.094 per sample. Three extracting errors all happened in the cases of fragile texts of invoice number as shown in Fig. 11.

Figure 12 shows some of the testing images and demonstrates the extraction result placed below. Figures 12(a)~12(d) are normal images, Fig.'s 12(e)~(h) shows the complex background cases which needs to use piecewise mapping function to expend the contrast. Figure 12(i)~12(l) are the detecting result of ill circumstances. The experimental results demonstrate that the proposed method is promising and is very robust to various complex situations.

5 Conclusions

This study develops an automatic system for detecting and extracting the invoice number on invoice image with various circumstances. The proposed procedure is divided into two main parts: invoice number detection and candidate invoice number segmentation. In the invoice number detection stage, the original image is first converted into grayscale image based on blue primary color. Next, fixed thresholding method removes the background. Then, complexity analysis and linear piecewise mapping function are applied to remove noise. The second stage, candidate invoice number segmentation, performs extraction of the invoice number region utilizing. morphology operation and RLSA which connects the object regions. Connected component is then adopted to group objects. Finally, we use false candidate removal algorithm to determine whether the candidate area is a real invoice number by inspecting the aspect ratio and the size of area. Experimental results indicate that the proposed approach can extract the invoice number automatically and achieves superior correct extraction performance up to 98.42%, while the false positive rate is as low as 0.094 per sample. The developed system is promising and is very robust to various complex situations.

Acknowledgement

This work is partially supported by Orbit Technology Inc.. The authors would like to thank Mr. Min-Chueh Lin for valuable comments.

Figure 12 Sample results of detection and extraction. (a)~(d) Images with clear background. (e)~(h) Images with complex background. (i)~(l) Image with various ill circumstances.

References

[1] Y. Zhong, K. Karu, and A. K. Jain, “Locating text in complex color images”, Pattern Recognition., Vol. 28, pp. 1523–1535, Oct. 1995.

[2] X. Gao and X. Tang et al., “Automatic news video caption extraction and recognition”, in Proc. LNCS 1983: 2nd Int. Conf. Intelligent Data Engineering. Automated Learning Data Mining, Financial Eng., Intell. Agents, K. S. Leung et al., Eds., Hong Kong, 2000, pp. 425–430.

[3] J. Ohya, A. Shio, and S. Akamatsu. “Recognizing characters in scene images”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(2):214–220, Feb. 1994.

[4] Y.M.Y. Hasan and L.J. Karam. “Morphological text extraction from images”, IEEE Transactions on Image Processing, 9(11):1978–1983, 2000.

[5] O. Hori. “A video text extraction method for character recognition”, In Proceedings of the International Conference on Document Analysis and Recognition, pages 25–28, 1999.

[6] A. K. Jain, B. Yu, “Automatic Text Location in Images and Video Frames”, Pattern Recognition, Vol. 31, No. 12, pp. 2055-2076, 1998

[7] .H. Kuwano, Y. Taniguchi, H. Arai, M. Mori, S.Kurakake, H. Kojima, “Telop-on-demand: Video Structuring and Retrieval Based on Text Recognition”, IEEE International Conference on Multimedia and Expo, pp. 759-762, New York, NY, July 2000.

[8] C. -M. Lee and A. Kankanhalli, “Automatic Extraction of Characters in Complex Scene Images”, International Journal of Pattern Recognition and Artificial Intelligence, Vol. 9, No. 1, pp. 67-82, 1995.

[9] H.K. Kim, “Efficient Automatic Text Location Method and Content-Based Indexing and Structuring of Video Database”, Journal of Visual Communication and Image Representation, Vol. 7, No. 4 pp. 336-344, Dec. 1996.

[10] A. Miene, Th. Hermes and G. Ioannidis, “Extracting Textual Inserts from Digital Videos”, International Conference on Document Analysis and Recognition, pp. 1079-1083, Sept. 2001.

[11] E. Ortacdag, B. Sankur, K. Sayood, “A New Algorithm in Locating Text in Complex Color Images”, International Journal on Document Analysis and Recognition, 1998.

[12] R. Lienhart, A. Wernicle, “Localizing and Segmenting Text in Images and Videos”, IEEE Transactions on Circuit Systems and Video Technology, Vol. 12, No. 4, pp. 256-268, 2002.

[13] V. Wu, R. Manmatha, E.M. Riseman, “Finding Text In Images”, Proc. of the 2nd ACM International Conference on Digital Libraries, R. B. Allen and E. Rasmussen, eds., pp.3-12, Jul. 1997.

[14] H. Makino. “Representation and segmentation of document images”, Proc. of IEEE Computer Society Conference on Pattern Recognition and Image Processing, 1983, pp. 291-296.

[15] D.T. Lin, M.C. Lin, K.Y. Huang, “Real-time Automatic Recognition of Omnidirectional Multiple Barcodes and DSP Implementation”, submitted to Computer Vision and Applications Journal.

[16] K.Y. Wang, R.G. Casey, F.M. Fahl, “Document analysis system”, IBM Journal of Research and Development, Vol. 26, No. 6, Nov 1982, pp. 647-656.

[17] H. L. Bai, J. M. Zhu and C. P. Liu, “A fast license plate extraction method on complex background”, in Proc. IEEE Intelligent Transportation Systems, Vol. 2, pp.985-987, Oct. 2003

[18] W.G.. Zhu, G. J. Hou and X. Jia, “A study of locating vehicle license plate based on color feature and mathematical morphology”, in Proc. IEEE International Conference on Signal Processing, Vol. 1, pp. 748-751, 2002

[19] D. Wang and S.N. Srihari. “Classification of newspaper image blocks using texture analysis. Computer Vision”, Graphics and Image Processing, 47:327–352, 1989.

[20] J. Kanai, M.S. Krishnamoorthy, T. Spencer Algorithm for Manipulating nested block represented images. SPSE’s 26th Fall Symposium, Arlington VA, USA, Oct 1986, pp.190-193.