Removal of Background Patterns and Signatures for Magnetic Ink Character Recognition of Checks

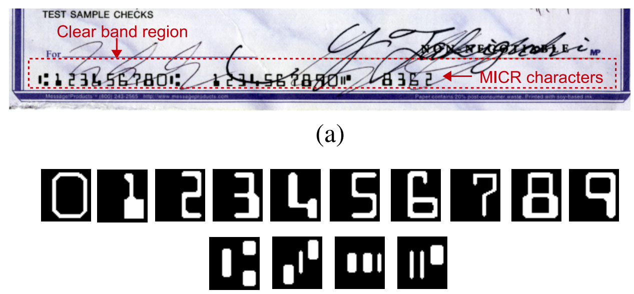
Keiichiro Shirai, Masashi Akita, Masayuki Okamoto
Faculty of Engineering, Shinshu Univ.
Nagano, Japan
{*shirai,okamoto*}*@cs.shinshu-u.ac.jp*

Kazuya Tanikawa, Takaaki Akiyama, Tetsuji Sakaguchi
SEIKO EPSON, Corp.
Nagano, Japan

***Abstract*—This paper describes a method to extract the magnetic ink characters (MICR E-13B font) printed on bank-checks for the purpose of using OCR as supporting MICR. In the case of OCR, the colorful background patterns and the overlapped signatures on MICR characters make it difficult to extract characters respectively by using simple binarization and labeling.**

Our method estimates the color and pitch of MICR characters in order to separate the characters in contact with sign strokes, then the remaining sign strokes are removed by tracing them. In the experiment, we use circulated bank-checks and samples provided by SEIKO EPSON and show the performance of our method.

***Keywords*-OCR; MICR; character separation; stroke tracing;**



(a)

(b)

Figure 1. (a) MICR characters printed on the clear band region (dashed rectangle) with degradations by colorful background and overlapped signatures and (b) templates of MICR character (E-13B font) consist of numbers $0-9$ and symbols $s_1 - s_4$.

I. Introduction

In automatic processing of bank-checks [1], several applications have been proposed such as a recognition of the amount of money [2], [3], information about customers [4] and the magnetic ink character recognition (MICR) [5], [6] which we mainly deal with in this paper. Originally the MICR characters are designed for the recognition using the distribution of magnetism as shown in Figure 1. However in recent years the OCR for MICR characters is becoming general by the performance improvement of OCR. In addition, higher accuracy recognition becomes important because the bank-checks can be dealt with only by electronic forms due to a recent law revision in the United States.

OCR methods [5], [6] and products [7], [8] for reading MICR characters have been proposed so far, and they work well for circulated bank-checks with good print conditions. But sometimes MICR characters are degraded by a colorful complex background or overlapped strokes of a signature, in such case the recognition accuracy of conventional methods drastically decrease. With the stroke fragments, some image features specialized for OCR like PDC[1] and P-LOVE[2] do not work well.

In this paper, we present a background and signature removal method as a preprocessing of OCR for MICR (E-13B) characters[3]. Our method consists of three components, (1) character color estimation and binarization to remove background, (2) character pitch estimation to separate each character, (3) sign fragments removal by stroke tracing. Considering a risk of failure in tracing strokes, and keeping characters from being deleted by mistaken for strokes at wide range, we first separate each character then remove fragments of strokes by tracing them from the boundary of a detected character region.

The rest of this paper is organized as follows. Sec. II describes the background removal by mean-shift clustering. Sec. III describes the character segmentation by pitch estimation. Sec. IV describes the signature removal by a stroke analysis, Sec. V shows experimental results.

II. Background removal by character color estimation

The color of MICR character is almost black, but occasionally the color is paled out due to the ink rejection. Additionally, since some bank-checks have dark colors, the threshold range of binarization has to be set narrowly and adaptively. To estimate the center of threshold range, we use mean-shift clustering and describe the way in this section.

[1]PDC: peripheral direction contributivity feature
[2]P-LOVE: peripheral local outline vector feature

[3]Although there are two types of fonts E-13B and CMC-7 in the MICR character, only E-13B is considered here for the convenience of patent [9].

978-0-7695-4661-2/12 $26.00 © 2012 IEEE
DOI 10.1109/DAS.2012.75

Figure 2. Binarization with estimated character color.

A. Ink color clustering by mean-shift in histgram domain

The mean-shift [10], [11] is known as a simple and fast nonparametric clustering, and is usually applied to the RGB color data. However we apply it to a histogram of each R,G,B color component independently $\{H_c\}_{c=r,g,b}$ because the ink colors distribute near black and along the axis toward black:

$$s_c^{(t+1)} = \sum_{b \in N(s_c^{(t)})} bH_c(b) / \sum_{b \in N(s_c^{(t)})} H_c(b), \qquad (1)$$

where $s_{c=r,g,b}$ is an ink seed color and it obtained by taking the mean of neighboring color (bin index) b, and $H_c(b)$ is the number of pixels having the color. After some iterations, the seed color converges to the ink color. In our experiment, we set the initial seed color as black $\mathbf{s}^{(0)} = [s_r = 0, s_g = 0, s_b = 0]$, the number of histogram bins as $\#b = 256$, the radius to search neighbors as $N(s_c) = [s_c - 3, s_c + 3]$ and the number of iterations as $t = 10$.

Using the obtained seed color $\hat{s}_{c=r,g,b}$, a check image I is binarized by use of constant margin $[-\tau, +\tau]$ from the seed color:

$$B(\mathbf{p}) = \begin{cases} 1 & \forall_{c=r,g,b}\ \hat{s}_c - \tau \le I_c(\mathbf{p}) \le \hat{s}_c + \tau \\ 0 & \text{otherwise} \end{cases}, \qquad (2)$$

where $\mathbf{p} = (x, y)$ is pixel coordinates, and pixels with color $I(\mathbf{p})$ inside of the seed-centered square are selected as character candidates. The margin is set as $\tau = 0.05$ against the whole color range $[0, 1]$.

Above mentioned color estimation and binarization are actually done at compartmentalized blocks of a clear band because the ink color changes slightly at each region. We roughly decide the initial clear band region with enough margins and divide the region into 4 blocks in a horizontal direction.

Figure 2 shows a part of a binarized image shown in Figure 1. In a case of this image, background patterns were completely removed, and the remaining strokes will be removed in the following process described in Sec. IV.

III. Character Segmentation by Pitch Estimation

When fragments of background or sign strokes are in contact with some characters, the large connected component has to be segmented into each character. We first obtain the 2D character likelihood map by correlation filtering with a kernel generated from template characters, then project the correlation onto 1D domain and enhance the peak positions so as to observe each character pitch as the interval of peaks.

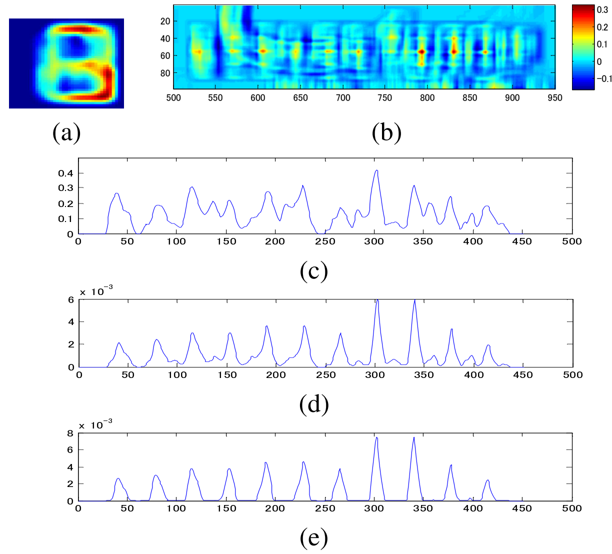


Figure 3. Likelihood estimation of character position. From top (a) bundled templates, (b) correlation image, (c) projected correlation (maximum value in vertical direction), (d) peak likelihood estimation, (e) dummy peak supression.

A. Character locating by bundled template-matching

Usually to know a particular character position by correlation filtering (phase correlation), the corresponding template image is used as the filter kernel. In contrast, we want to know roughly all position of characters at once, thus use a bundled template image $\bar{T}$ as shown in Figure 3a which consists of weighted sum of all templates $\{T_k\}_{k=\{0-9,s_1-s_4\}}$ (Fig. 1b):

$$\bar{T} = \frac{1}{K} \sum_k w_k T_k, \qquad (3)$$

where w_k is a weight to equalize occurrences of template strokes. Each weight is set as $w_k = \{2, 2, 3\}$ for $k = \{0, 4, 7\}$ templates and $w_k = 1$ to others. K is normalization term so that $\sum_{\mathbf{q} \in \Omega} \bar{T}(\mathbf{q}) = 1$ in the template region Ω. However, since this filter also responds to all-white region, we use a differential version of the filter:

$$\bar{T}' = \bar{T} - \#\Omega^{-1}, \qquad (4)$$

where $\#\Omega$ is the number of pixels in Ω and $\#\Omega^{-1}$ indicates the mean value of $\bar{T}$.

By using the above filter kernel $\bar{T}$, the correlation image $C_{2\mathrm{D}}$ is give as

$$C_{2\mathrm{D}} = \bar{T}' * B, \qquad (5)$$

where $*$ is the correlation operator. Figure 3b shows a correlation map $C_{2\mathrm{D}}$.

B. Likelihood of peak position estimation in 1D

Character pitches are mainly estimated from peak positions in a horizontal direction. To reduce the dimension, we take maximum values in each columns of the 2D map and obtain 1D correlation map C_{1D} (Fig. 3c). Although these peak positions are almost corresponding to the correct location of characters, dummy peaks also will appear. Furthermore the large variation of peak values makes it difficult to define the binary positions by a simple thresholding.

In order to enhance the peak positions (Fig. 3d), we consider the peak likelihood given from neighboring peaks. Ideally, MICR characters are printed with the same pitch d and the neighboring peaks appear at a distance d from a peak x with existing probability. The likelihood at x given from neighboring positions $x_d = x + d$ within the range $[-\ell, +\ell]$ is defined as follows:

$$\mathcal{L}(x) = \int_{-\ell}^{\ell} C_{\mathrm{1D}}(x_d + \ell)\mathcal{N}(x_d + \ell; \mu = x_d, \sigma)d\ell, \quad (6)$$

where $\mathcal{N}(\cdot)$ is the existing probability as a Gaussian distribution with the origin position x_d and standard deviation σ. Then, using the likelihood $\mathcal{L}$, the current correlation map is enhanced by multiplying each other:

$$C'_{\mathrm{1D}}(x) = C_{\mathrm{1D}}(x) \cdot max(\underset{x_d=x-d}{\mathcal{L}(x)} , \underset{x_d=x+d}{\mathcal{L}(x)}). \quad (7)$$

Note that actually two likelihood are calculated for the both side peaks $x_d = x \pm d$ and take the larger value, this is a way to deal with the left-most or right-most character which appears with only one neighboring peak. In the experiment, the default pitch is set to around $d = 40$ and the standard deviation is set to $\sigma = 2.5$, ℓ is set to 4σ.

C. Dummy peak suppression

MICR characters have many vertical line segments, and the sequence of them sometime make character-like patterns at the position of half pitch $x_{d/2} = x + \frac{d}{2}$, as a result the correlations increase wrongly.

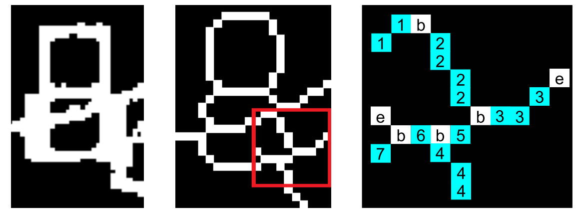
The magnitude of dummy peaks is generally smaller than that of true peaks, thus we compare the correlation of a position x and $x_{d/2}$ and enlarge the larger peak, in contrast suppress the smaller one. In practice, we use $x_{d/2}$ instead of x_d in Eq. (6) and utilize the following sigmoid function instead of the function $\mathcal{N}(\cdot)$:

$$\mathcal{S}(\Delta c; \rho) = \frac{1}{1 + e^{-\rho \Delta c}}, \quad (8)$$

where $\Delta c = C'_{\mathrm{1D}}(x) - C'_{\mathrm{1D}}(x_{d/2})$. The gain coefficient is set to $\rho = 0.5$ in the experiment. Figure 3e shows the correlation distributions after suppressing dummy peaks.

D. Determination of peak positions

Final binary peak positions are extracted by adaptive thresholding. As the thresholds, a smoothed correlations by a Gaussian filter $\mathcal{N}(\mu = 0, \sigma = 1)$ is used. Then, in each extracted region, a position having maximum value is selected as a candidate.



(a) (b) (c)

Figure 4. Sign stroke tracing from the boundary of cropped region. (a) Input banalized character and stroke fragments, (b) Resulting image of thinning. (c) Stroke components analysis: endpoint 'e', branch 'b' and labeled line segments.

To remove the dummy peak completely, the value of candidates are compared again. In a case there is another peak within half pitch of a peak, smaller peak is regarded as a dummy peak and removed.

Finally, regions with the certain margin to the left and right from each peak are extracted as character regions (Fig. 8).

IV. Signature removal by stroke analysis

Fragments of remaining sign strokes in separated character regions can be removed by stroke analysis [12]. In our method, generally sign strokes come from the outside of the boundary of a character region and they tend to be in contact with the boundary (Fig. 4a). Therefore we trace the sign strokes from boundary toward inside, considering the curvature of thinned skeleton of signs and a character.

A. Thinning and structure of line segments

As the thinning method, we use [13] which is also implemented on MATLAB, and apply a 3×3 median filter in advance so as to get smoother lines of skeletons. Figure 4b shows the result of thinning.

Then pixels in the skeleton are grouped into the following three groups associated with the connectivity number $\#c$:

$$\begin{cases} \text{End point} & \#c = 1 \\ \text{Line element} & \#c = 2 \; . \\ \text{Branch point} & \#c \geq 3 \end{cases} \quad (9)$$

Figure 4c shows the partially enlarged grouped pixels. From each end point, each pixel is traced until reaching a branch point, and they are grouped into line segments.

B. Stroke tracing by considering curvature

Sign strokes are traced by the following procedures:

(i) Select an end point touching the boundary as a start point. Trace a line segment until reaching a branch point.

(ii) At the branch point, find all neighboring line segments, then calculate a curvature between a new line segment and the already traced line.

(a) (b) (c)

Figure 5. Sign stroke removal and character shape recovery. (a) Detected sign strokes. (b) Sign stroke removal with dilated stroke. (c) Character shape recovery by dilated branch points.

1 4 5 7 s_2 s_3

Figure 6. Failure examples of the stroke removal. A sign stroke overlap an edge of character in the same direction.

(iii) Continue tracing toward a line with the smallest curvature.

(iv) Repeat (ii)-(iii) until reaching another end point.

If the length of a line segment is not long enough to calculate the curvature such as 5th and 6th segments in Figure 4c, these segments are merged with an adjacent segment, then the curvature of them is calculated.

The curvature of strokes used in (ii) is calculated as:

$$k(t) = \frac{x_t y_{tt} - y_t x_{tt}}{(x_t^2 + y_t^2)^{3/2}}, \tag{10}$$

where subscript t indicates the derivatives of coordinates $(x(t), y(t))$. The first and second order derivatives are given as:

$$\begin{array}{l} x_t = \frac{1}{\Delta t}\{x(t - \frac{\Delta t}{2}) - x(t + \frac{\Delta t}{2})\} \\ x_{tt} = \frac{1}{\Delta t^2}\{x(t - \Delta t) - 2x(t) + x(t + \Delta t)\} \end{array}, \tag{11}$$

also y_t, y_{tt} are given in the same manner, and the number of samples for calculation is $\Delta t = 4$ in our experiment. Figure 5a shows an example of extracted sign strokes.

C. Sign stroke removal and character shape recovery

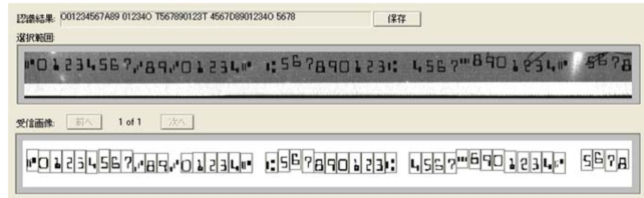
Pixels corresponding to the sign strokes are removed by a morphological dilation operation (Fig. 5b, colored pixels). However this operation will also carve the shape of a character, in order to recover the shape, we use dilated branch points to recover the loss. Figure 5c shows the result of interpolation, and this result is used for the OCR process.

V. Experimental results

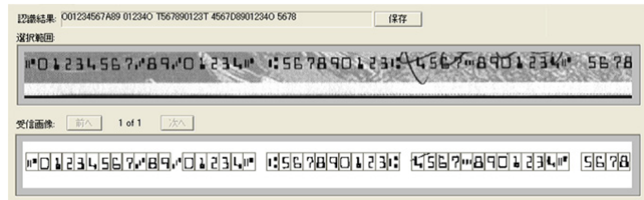
In the experiment, we use 1449 bank-checks scanned with 300dpi color including circulated checks and samples. The samples have considerable degradations by dark background and strong edges, thin character color by ink reflection, wavy position of characters and signature overlapping. As conventional methods, we use SDK products [7], [8] which have a special OCR function for MICR characters. The resulting images are shown in Figure 7, from the left, (a) original image, (b) segmentation result described in Sec. III, (c) our method and (d),(e) show the conventional methods. In our method shown here, we use our original OCR engine for the final recognition.

For the images (a),(b), backgrounds and signatures were removed and the shape of characters were recovered successfully. The result (c) shows an example of wavy characters, in such a case, a character touching the boundary is infrequently removed by mistaken for strokes. Meanwhile, a failure case is shown in (d), our method is not good at treating a overlapped sign stroke on edges of a character in the same direction like the 5th character. Figure 6 shows the failure results of stroke removal, these characters are used for OCR directly therefore the accuracy depends on the image features used in OCR engine.

Figure 8 shows the actual simulation result with a live test data set provided by SEIKO EPSON [9]. Although the final recognition accuracy is depend on the OCR engine, our method helps to remove unnecessary objects and separate characters.



(a) Degradation by wavy character positions

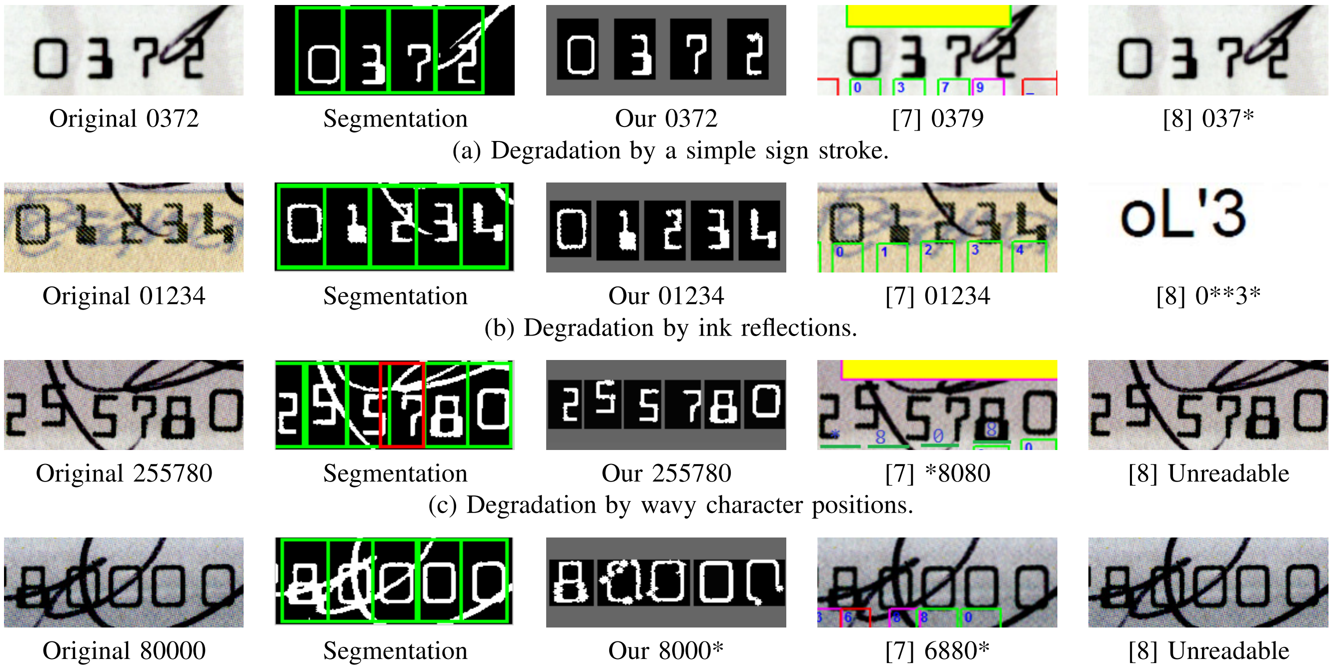


(b) Degradation by a complex background

Figure 8. OCR for MICR simulation result provided by SEIKO EPSON Corp. [9]. The 1st row's text box shows the recognition result, 2nd and 3rd rows show the input image and its segmentation result.

VI. Conclusions

An OCR based MICR 13-B character extraction method is described in this paper. We mainly consider how to deal with the degradation of characters by complex backgrounds and signatures. To further improve the recognition accuracy and the character separation accuracy, a hybrid recognition using image features of OCR and pure MICR bilaterally is required. Also the adaptive parameter setting instead of using heuristic parameters is a task.



(a) Degradation by a simple sign stroke.
(b) Degradation by ink reflections.
(c) Degradation by wavy character positions.
(d) Degradation by heavy sign strokes.

Figure 7. Experimental results of our method and conventional methods [7], [8] for comparison. The images of conventional methods are raw outputs at the same position.

ACKNOWLEDGMENT

This work is a part of collaborative research with SEIKO EPSON conducted within 2008-2010 [9]. The shown algorithm in this paper is our original version, and the modified version is implemented on the product so that the algorithm works with reasonable processing speed and accuracy.

REFERENCES

[1] S. Impedovo, P. S. P. Wang, and H. Bunke, *Automatic Bankcheck Processing*. World Scientific Pub Co Inc, 1997.

[2] A. Agarwal, L. Granowetter, K. Hussein, and A. Gupata, “Detection of courtesy amount block on bank checks,” in *Proc. of Inter. Conf. on Document Analysis and Recognition*, 1995, pp. 748–751.

[3] N. Gorski, V. Anisimov, E. Augustin, O. Baret, D. Price, and J.-C. Simon, “A2ia check reader: A family of bank check recognition system,” in *Proc. of Inter. Conf. on Document Analysis and Recognition*, 1999, pp. 523–526.

[4] S. Madhvanath, S. McCauliff, and K. M. Mohinddin, “Extracting patron data from check images,” in *Proc. of Inter. Conf. on Document Analysis and Recognition*, 1999, pp. 519–522.

[5] F. Chin and F. Wu, “A microprocessor-based optical character recognition check reader,” in *Proc. of Inter. Conf. on Document Analysis and Recognition*, vol. 2, 1995, pp. 982–985.

[6] G. Dimauro, S. lmpedouvo, G. Pirlo, and A. Salzo, “Automatic bankcheck processing : A new engineered system,” *Inter. Journal of Pattern Recognition and Artificial Intelligence*, pp. 467–504, 1997.

[7] Inlite, “Clearcheck21 micr reader,” http://www.inliteresearch.com/.

[8] LEADTOOLS, “The world leader in imaging sdks,” http://www.leadtools.com/.

[9] SEIKO EPSON, Corp., “Removal of background patterns and signatures for magnetic ink character recognition of checks,” IPC 2010-191724, 2010.

[10] Y. Cheng, “Mean shift, mode seeking, and clustering,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790–799, 1995.

[11] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[12] M. Suwa, “Segmentation of connected handwritten numerals by graph representation,” in *Proc. of Inter. Conf. on Document Analysis and Recognition*, vol. 2, 2005, pp. 750–754.

[13] L. Lam, S.-W. Lee, and C. Y. Suen, “Thinning methodologies-a comprehensive survey,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 14, no. 9, pp. 879–885, 1992.