Original research article

Handwritten Student Identification Number Segmentation on Traditional Grid Answer Sheet Based on Statistical Analysis of Structural Profile

Nutchanat Sattayakawee*
Department of Computer Engineering, Faculty of Engineering,
Kasetsart University Kamphaengsaen Campus, Nakhon Pathom 73140, Thailand

Received 30 August 2016; Received in revised form 1 November 2016
Accepted 10 November 2016; Available online 24 March 2017

ABSTRACT

As part of an attempt to improve computer-based test scoring on traditional grid answer sheets, which are used by 76.36% of interrogated schools in Thailand, this paper proposes an algorithm to segment the problematic handwritten-style student identification number. The method used is based on profile features supported by digit width-height ratio analysis. Non-touching digits are segmented by vertical histograms and the flood fill algorithm. Touching digits are segmented by considering the distances between upper and lower profiles as well as the derivatives of consecutive points of their profiles. Among 231 six-digit image sequences, the average segmentation accuracy rate was 98.20%, which is considered satisfactory. The results showed that an increased number of consecutive cursive digits decreased the segmentation accuracy, and was significant from 5 connecting digits.

Keywords: Student identification number; Handwritten digit segmentation; Profile; Flood fill

Introduction

It is undeniable that multiple-choice tests are commonly utilized to evaluate students in most schools in Thailand. Although automatic Optical Mark Recognition (OMR) solutions have been commercially available in Thailand for years, they have not been commonly used. Only 26 out of 110 interrogated secondary schools in Thailand have them according to surveys of the usage of OMR in 2012 [1] and 2015. In contrast, the majority of schools have been using traditional grid answer sheets as shown in Fig. 1. The answer sheet requires instructors to grade their students manually, which is labor intensive and time consuming. Therefore, developing computer-based test scoring with a low-cost answer sheet is our main objective.

Grading tasks consist of scoring the correct answers marked on the sheet and identifying the student who owns the answer sheet. For computer-based processing, the first step involves mark recognition in the answer grid. This task was achieved with an accuracy of 99.91% [1]. The next step is to make character recognition of student information, which is presented at the top of the answer sheet. However, extracting student information from the form will make

***Corresponding author:** fengncn@ku.ac.th

the task particularly complicated since the information comes from either the combination of more than one separated fields in the form, e.g. class and number, or the recognition of Thai characters. To simplify the problem, we proposed letting the student write his or her student identification number (SIDN) at the top-left corner of the answer sheet [2]. Consequently, Thai character recognition is substituted with digit recognition. Nevertheless, the accuracy rate of 77.50% for digit recognition still needs to be improved [2].

The main processes of Optical Character Recognition (OCR) are image pre-processing, image segmentation and image recognition. From our analysis, image segmentation in previous work [2] can only be applied to well-written digits consisting of a continued stroke for each digit and simple two touching digits. However, this may not consistently happen. Therefore, this research has focused on image segmentation of digits on traditional grid answer sheets in order to improve computer-based test grading.

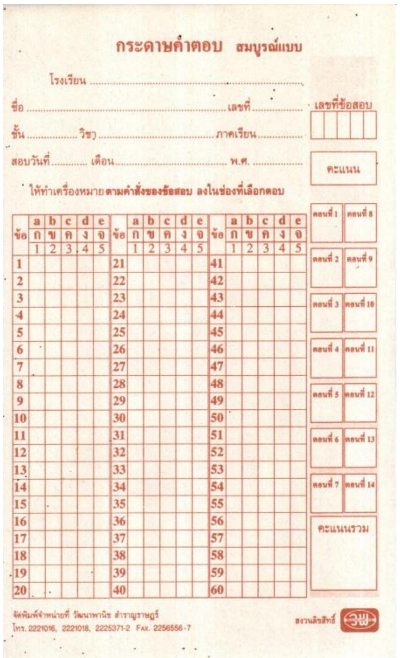


Fig. 1. Example of a traditional grid answer sheet in Thailand.

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

Fig. 2. Different problem types for digit segmentation: (a) touching, (b) overlapping, (c) surplus stroke connection, (d) broken, and (e) slanted digits.

The following sections are organized as follows. An overview of handwritten digit segmentation is in section 2. Section 3 explains the source of our dataset and gives details of our proposal. The results and conclusion are presented in sections 4 and 5, respectively.

Handwritten Digit Segmentation

Digit segmentation is the process of isolating each digit into a sub-sequence, given a single sequence of digits. It is one of the main challenges in the OCR problem; however, some recognition algorithms bypass this step [3]. These can be called segmentation-free methods. Segmentation can be classified into two types: explicit and implicit [4-5]. Explicit segmentation means that candidate characters are segmented a priori and provided to the recognizer afterwards. Implicit segmentation means the process is done simultaneously with recognition. Explicit segmentation seemed to achieve better results [4]; thus, explicit segmentation is our main focus.

Segmentation problems

Handwritten digit segmentation problems arise from the fact that the sequence of digits can be written in an unconstrained way. Most handwritten digit segmentation algorithms tried to solve two or more cursive digits or sometimes overlapping ones [4-7], as illustrated in Fig. 2(a) and Fig. 2(b). Two-touching numerals, which represent 80% of all touching components [8], are the most common. Connected digits can also emerge from the running hand making an additional stroke on the hand's motion up connecting two or more digits as shown in Fig. 2(c). This type of connection is less mentioned than others; however, such a useless stroke can be eliminated by the method proposed by Elnagar and Alhajj [9]. Broken digits as shown in Fig. 2(d) can also cause the segmentation to fail. This problem can be solved by reconstructing the digits, i.e. filling in discontinued strokes, or by just clustering back all the separated segments of the same digit into one segment [10-11]. Illustrated in Fig. 2(e), slanted digits can also pose some problems in segmentation and recognition. This can be corrected by the slant removal algorithm [12]. However, sometimes, in spite of the existence of touching skewed digits, segmentation can be done using skeleton analysis or oriented sliding windows [13-14].

In the proposed algorithm, we have focused on slightly slanted or upright touching digits (as in Fig. 2(a) and 2(c)), broken digits (as in Fig. 2(d)) and non-cursive slanted digits (as in Fig. 2(e)). Furthermore, our techniques, in particular for broken and slanted digits, are different from the mentioned literatures.

Segmentation features

All algorithms are initially defined based on segmentation features, which influence not only subsequent steps of the process but also the pre-processing one. The latter includes smoothing, thinning, normalization, slant correction, etc. The features the most adopted are profile, contour and skeleton analysis [2, 5, 6, 9, 13, 14, 15, 16]. Some methods are based on a single feature [2, 5, 6, 9, 15], others on multiple features [13, 14, 16].

Isolated or touching digits can be first classified by a vertical projection profile. Furthermore, the distance between lower and upper profiles can be used to determine touching zones of digits. The upper profile is formed by the distances between the upper boundary and the first foreground pixel in each column while the last foreground pixel is used instead for the lower profile. Image contour is like an envelope of the image. By considering some contour points, segmentation points can be determined [13, 15]. A skeleton of the image can be processed by the thinning algorithm, making the stroke of the image one pixel wide while preserving its structure. It is generally used to find the junctions of the strokes, which have potential to be segmentation points [5, 6, 9, 13].

Subsequently, the concept of a water “reservoir”, a large space between touching digits, was proposed for finding touching points [17]. Segmentation is done by using the structure of the reservoirs and analyzing their sizes and their heights. Another interesting approach is graph representation of touching patterns of digits [16]. Matrices relating to the connected graph are calculated and segmentation is then performed by applying graph theory and heuristic rules.

Based on these features, each algorithm was achieved by combining morphological analysis, neural network, heuristics, or other different methods.

Materials and Methods

Image samples

Although there are several standard databases of handwritten digits for segmentation and recognition [4], the main purpose of this research is not only to process image segmentation, but also student identification from the answer sheets. As a result, the source of image samples should be selected from writing on traditional grid answer sheets because pre-processing of low-quality paper should be done to find the image region of the SIDN.

Image samples collected from a school were used in our previous study [2]. However, the way the students wrote their SIDNs was so neat that segmentation could be done easily. This may not be the case when the students are taking the actual examination.

Consequently, the samples mainly studied in this work were newly collected from 118 students at the Faculty of

Engineering at Kamphaengsaen, Kasetsart University. Each student wrote his or her six-digit number twice, once under normal conditions and another in a rush. SIDNs were written in pencil or blue or black ballpoint pen at the top-left corner of the answer sheets. The sample images were acquired using an auto-feed scanner with 300 dpi resolution. After eliminating inadequate writing samples, which did not contain six-digit SIDNs, there were a total of 231 usable images.

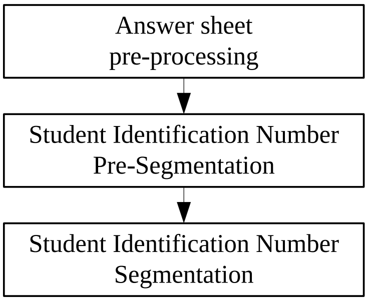


Fig. 3. Flowchart of SIDN segmentation on traditional grid answer sheet.

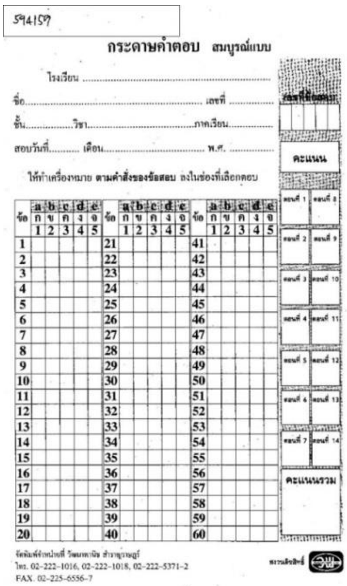


Fig. 4. SIDN area in the answer sheet.

Proposed method

The proposed method tries to improve the segmentation process in our previous work [2] and is still based on structural analysis. A correlated structure of characters of a person's writing leads to the proposed technique of cursive digit segmentation which is the main segmentation problem. Furthermore, in our opinion, segmentation does not need to be perfect as long as the segmented digits can still be recognized by humans.

The overall structure of numeral segmentation is shown in Fig 3. There are three main steps which are described below.

Answer sheet pre-processing

Image pre-processing is performed in a manner similar to the one utilized in our previous work [2] and consists of image cropping, binarization and median smoothing. First 50% of the width and 5.3% of the height of the answer sheet image is cropped from the top left corner. The resulting image is the top left rectangle area in Fig. 4. This region is supposed to contain the SIDN whose length and height will not exceed that region, considering normal writing of 0.3 – 0.5 cm high characters. Previously, colored images were converted to gray scale and binary ones, using a specified threshold [2]. Much noise from the background still remained in the cropped zone using this method. As a result, the binarization process should be improved.

The color of the line and text of the answer sheet is RGB (231,141,114), which is a dark orange tone. Moreover, the background color is light cream with about RGB (249,249,237). From experimentation, it was found that YCbCr color space and Otsu's threshold [18] could be used to separate the foreground and the background while cleaning up noise pixels very well. Nevertheless, the stroke of the foreground was also expanded, sometimes making originally non-touching digits become touching ones. Analyzing the colors of the background and foreground, the threshold of 200 in only the R channel was opted to separate background and foreground as shown in Fig. 5. This figure presents the average of R values of student number zones from a random one-third of image samples. The reason why this value was selected is because from the value 255 down, the frequencies of R values become roughly constant below the value 210 before slightly increasing from 170 down. We can see that the frequencies are very high when R values are above 240 because most of the area is the background. After that, median smooth and noise reduction are applied to the image. An example image of this step is illustrated in Fig. 7(b).

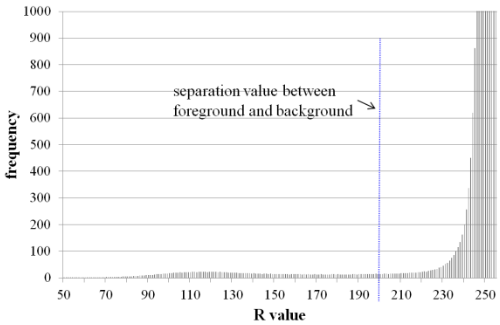


Fig. 5. R channel frequency for image binarization threshold.

SIDN pre-segmentation

After the first step, sometimes some noise still remains on the binary image due to non-uniform background color of the answer sheet; moreover, large white space around the digits still exists. This step is concerned with cropping the image to get only the sequence of digits area as in Fig. 8. This is done by gradually trimming the image edge where black pixels are sparse using vertical and horizontal projection profile.

SIDN segmentation

The aim of this step is to get sub-sequences of isolated digits. The assumption under this approach was that each person had his particular style to write characters, but differed among one another. Some may write high-thin characters, others may write wide characters, but they rarely change from one way to another when writing characters at once. Observing this, the characters of the handwriting of a person are correlated. This leads to the procedure we use to determine and segment cursive digits. Yet, the desired number of digits in the sequence, noted as DD, must be specified and is configurable in this proposed method. This process is summarized in Fig. 6 and detailed in the following sub-sections.

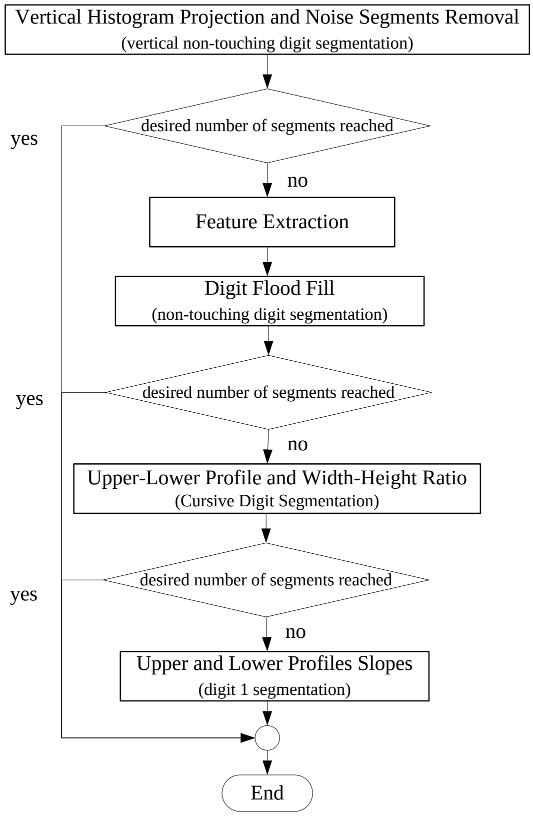


Fig. 6. Flowchart of SIDN segmentation.

Moreover, in order to show the meaning and the continuation of each step, only one example, which can illustrate all the steps, is selected. This latter is the sequence 251753 as shown in Fig. 7(b) although it may be recognized as 2157153 at first glance. This can be clearly seen in the colored image as shown in Fig. 7(a) where the upper horizontal stroke of digit 5 is extended to the front. We can see that the weight of those front strokes of digits 5 is lighter than the one of the real digit 1.

1) Vertical histogram projection and noise segments removal

Vertical histograms help isolate non-touching digit components. The resulting sub-images are then analyzed to check if they are noise segments or candidate digit segments by considering the width and height of each segment. Noise segments are identified when they are 30% below the width and height thresholds formed by the average and standard deviation (S.D.) of the widths and heights of all obtained segments. Detected noises are then removed.

(a)

(b)

Fig. 7. Sample image from answer sheet pre-processing: (a) in RGB, and (b) in black and white.

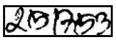


Fig. 8. Sample image from pre-segmentation.

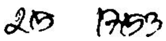
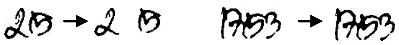


Fig. 9. Result from vertical projection segmentation of Fig. 8.



(a) (b)

Fig. 10. Digit flood fill of Fig. 9: (a) successful case, and (b) unapplicable case.

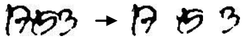


Fig. 11. Cursive digit segmentation of Fig. 10(b).

Fig. 12. Segmentation from analysis of upper and lower profiles slopes.

Fig. 13 shows a vertical histogram of Fig. 8. We can see that no black pixel is presented from positions 64 to 68. The single sequence is then divided into two sub-sequences presented in Fig. 9 after vertical segmentation has been performed.

2) Feature extraction

After eliminating the noise segments, if the number of sub-images is equal to DD, the process of segmentation is finished. If there are a greater than expected number of sub-images, there must be big noise segments. Fewer number of sub-images indicates touching digits. In both cases, simple correlation features of the digits are extracted. These features as well as their notations are listed below:

- W:= width per digit in a segment;
- W_{avg}, $W_{S.D.}$, W_{min}:= average, S.D. and minimum of W of all segments respectively;
- H:= height per digit in a segment;
- H_{avg}:= average of H of all segments;
- WHR:= width-height ratio of a digit, calculated by W/H;
- WHR_{avg}, $WHR_{S.D.}$, WHR_{min}:= average, S.D. and minimum of WHR of all segments respectively;
- EXPD:= expected number of digits in a segment, calculated from the rounded value of the segment width divided by its W;
- RDis:= difference between WHR and WHR_{avg} of a segment.

RDis is collected only if the EXPD of the segment is more than one. Furthermore, if the sum of EXPD of all segments exceeds DD, the EXPD of the segment, whose floating point value of expected digits is the furthest from its EXPD, is decremented by one. This is repeated until the sum is not above DD. An illustration of some features extracted in the case of Fig. 9, which contains two segments, is shown in Fig. 14, where DD is set to be six and the sum of EXPDs is only five.

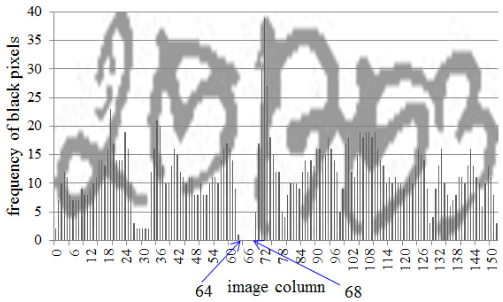
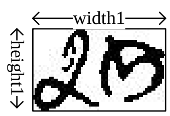
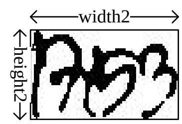


Fig. 13. Vertical histogram projection demonstration.



W_{avg} = (width1+width2)/DD
EXPD = [width1/ W_{avg}] = 2
W = width1/EXPD
H = height1
WHR = W/H

(a)



EXPD = [width2/W_{avg}] = 3
W = width2/EXPD
H = height2
WHR = W/H

(b)

Fig. 14. Some extracted features of Fig. 9.

3) Digit flood fill segmentation

The flood fill algorithm is applied after the segments with zero EXPD are removed. This procedure will only take action on the segments reported to have more than one EXPD. Segmentation by flood fill is done by following 8-direction consecutive black pixels. Black pixels are copied to a new image and replaced by the white ones until no consecutive black pixel is found. This can isolate the digits that are vertically overlapping but not touching, especially for slanted digits as shown in Fig. 2(e).

In fact, once flood filling is done, the size and black traits of the flood filled area are also evaluated in order to remove it if it is considered a noisy area. The removal occurs when black pixels of the newly flood filled area are less than 5% of black pixels of the previous segment. This is illustrated in Fig. 10(a) where sub-sequence 25 is isolated into segments of digit 2 and digit 5 while the broken head of digit 2 is removed by flood filling because the change is less than 5% and is considered to have no effect on the digit. However, the sub-sequence 1753 is not segmented as illustrated in Fig. 10(b) because removing the tail of digit 3 has a considerable effect. This method can deal well with broken digits.

If the segment contains only connected digits, flood filling has no effect. Hence, segmentation of cursive digits is performed thereafter.

4) Cursive digit segmentation

This is done by using the upper and lower vertical profiles. The range of positions of segmentation is pre-calculated from WHR_{avg} and $WHR_{S.D.}$ according to Eq. (1) – Eq. (4). Eq. (1) and Eq. (2) give the target positions P_{t1} and P_{t2} around which segmentation will occur. In Eq. (2) width is the width of an entire segment. Eq. (3) and Eq. (4) give lower and upper positions along the horizontal axis for each estimated sub-segment i, noted as P_{l_i} and P_{u_i}, where P_{begin_i} is the beginning position of the sub-segment, starting from 0 and incremented by W_{avg}.

$$P_{t1} = \left[\frac{WHR_{min} + WHR_{avg} + WHR_{S.D.}}{2} * H \right] \quad (1)$$

$$P_{t2} = \left[\frac{width}{EXPD} \right] \quad (2)$$

$$P_{l_i} = \min(P_{t1}, P_{t2}) - W_{S.D.} + P_{begin_i} \quad (3)$$

$$P_{u_i} = \max(P_{t1}, P_{t2}) + W_{S.D.} + P_{begin_i} \quad (4)$$

$$d_{i,j} = \left| l_{i,j} - u_{i,j} \right| \quad (5)$$

In a given sub-segment i, the distance between upper and lower profiles for each position j, noted as $d_{i,j}$, is calculated according to Eq. (5), where $u_{i,j}$ is the upper profile and $l_{i,j}$ is the lower profile in a sub-segment i at column j. The three positions with the shortest distance between the upper and lower profiles are considered to be the candidate positions, noted as C. The selected position, P_{sel_i}, is the one which meets the following conditions:

- $P_{sel_i} = \underset{j \in C}{\arg\min}(d_{i,j} + |j - (P_{begin_i} + P_{t2})|)$;
- $\exists(m,n):(m < P_{sel_i} < n) \wedge (d_{i,m} > d_{i,P_{sel_i}}) \wedge (d_{i,n} > d_{i,P_{sel_i}})$;
- $WHR_{avg} - WHR_{S.D.} \le WHR_i \le WHR_{avg} + WHR_{S.D.}$.

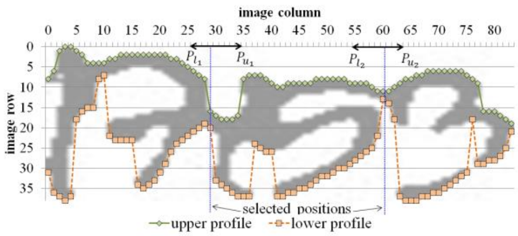


Fig. 15. Upper-lower profile using with width-height ratio demonstration.

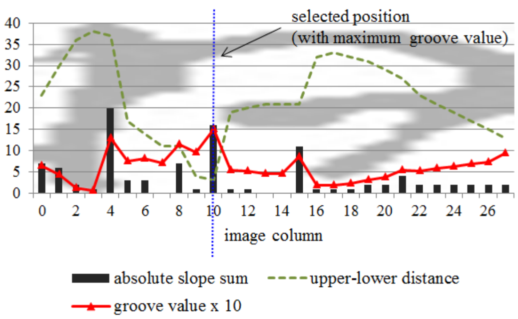


Fig. 16. Profile slope analysis demonstration.

If this is the case, the input segment is cut into two separated segments, or the segmentation is rejected. Each time the segmentation is done, the features of the segments are re-calculated. This process is repeated until the EXPD of all segments is one. Presenting the sequence with an EXPD of 3, Fig. 15 shows the selected positions and the ranges of candidate segmentation positions from this methodology. Its result is illustrated in Fig. 11.

5) Segmentation based on first order derivatives of profiles

It can happen that after the cursive digit segmentation phase, the number of segments is still less than DD. In this case, the writer usually wrote some touching numerals in a non-uniform way, i.e. WHRs of some written digits are too small. Profile slope analysis is processed for each segment currently obtained. Connecting digits in a segment normally contain grooves, i.e. peaks or valleys, from upper or lower profiles. This is detected by calculating the derivatives of both profiles. The segmentation position is chosen from slope analysis as well as upper-lower distance analysis. Eq. (6) calculates the slope sum (ss) of upper and lower slopes, denoted as s_u and s_l respectively, at position j of the image segment. Eq. (7) shows the calculation of groove value (GV) for each position j, where d_j is the distance calculated by Eq. (5) and d_{max} is the maximum of distances.

$$ss_j = |s_{u_j}| + |s_{l_j}| \tag{6}$$

$$GV_j = \frac{ss_j}{\max\limits_{0 \le j < width}(ss_j)} + \frac{d_{max} - (d_j + d_{j+1})/2}{d_{max}} \tag{7}$$

Segmentation position P_{sel} must meet the following conditions:

- $P_{sel} = \underset{0 \le j < width}{\arg\max}(GV_j)$;
- $d_{P_{sel}} < 0.6 * d_{max}$;
- The heights of two sub-segments are greater than $H_{avg}/2$.

Fig. 16 graphically demonstrates slope analysis of upper and lower profiles, which results in Fig. 12. From this example, the segment 17 was isolated into segments of digit 1 and digit 7. It is notable that this step is used to segment digit 1 since it consumes less area than other digits.

Finally, if the segmentation position cannot be identified from any segment, the digit segments are returned resulting in incomplete segmentation.

Results and Discussion

The first image set used in this experiment was previously studied [2]. The result of this proposed algorithm gave the segmentation rate of 100%, while 95.71% had been reported previously [2]. This resulted from the improvement of binarization method. Moreover, broken digits that our previous study [2] ignored were taken into account in this work.

As explained previously, our study analyzed 231 digit sequence samples. Each sample contained six digits. Consequently, there were a total of 1,386 digits. For each isolated sub-sequence of digits in any sample, we classified the digits into six types: single digits, two cursive (2C) digits, three cursive (3C) digits, four cursive (4C) digits, five cursive (5C) digits and six cursive (6C) digits. The numbers of sub-sequences in each group are shown in Table 1. All cursive-digit sequences made up 252 digits, which represented only 22.22%. Moreover, among connected digits, two-cursive-digit sequences were the most common, representing 74.04%, which was comparable to the percentage observed by Wang et al. [8]. Yet, in the tested image set, there were 14 broken digits, eight of which were digit 5 and the rest were other digits. These were all correctly detected and segmented. Some examples of broken digits are shown in Fig. 17. Correct and incorrect segmentation is illustrated in Fig. 18 and Fig. 19, respectively. Segmentation rates of each category are reported in Table 2. These percentages were calculated according to the correctness of each single segmented component. For instance, in Fig. 19(d), four-digit sequence segmentation gave the first three digits correctly and one incorrectly. The highest percentage was certainly the single digit type with 99.91% accuracy. The errors found in the single and 2C segment groups usually came from a non-uniform width-height ratio between the incorrect digits and the others in the same sequence. Segmentation accuracy rates dropped significantly in types 5C and 6C since the sequences were very complex. In addition, this might be due to the fact that there were few cases of these types, so arising errors greatly affected the accuracy rates.

Fig. 17. Examples of broken digits.

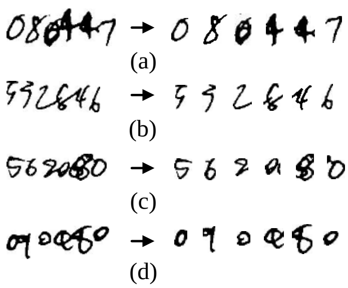


Fig. 18. Examples of successful segmentation.

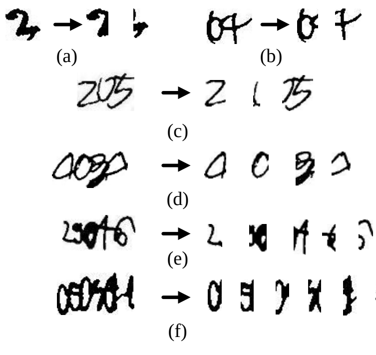


Fig. 19. Examples of error cases separated by types: (a) single, (b) 2C, (c) 3C, (d) 4C, (e) 5C, and (f) 6C.

The researches for segmentation of more than 2 touching digits were less contributed in this problem domain, as reported only 5 out of 15 reviewed methods by Ribas et al. [4]. Nonetheless, our performance in types 3C and 4C were 85.42% and 92.86%, respectively, which were in the same range as other algorithms [4]. This segmentation method gave a combined segmentation accuracy rate of 90.48% for touching digits and an overall average segmentation accuracy rate of 98.20%.

Table 1. Frequency of digit sequences in 6 types.

Single	2C	3C	4C	5C	6C
1,134	77	16	7	2	2

Table 2. Accuracy results of SIDN segmentation.

Type	Number of digits	Number of correctly segmented digits	Percentage
Single	1,134	1,133	99.91
2C	154	149	96.75
3C	48	41	85.42
4C	28	26	92.86
5C	10	7	70.00
6C	12	5	41.67
Total	**1,386**	**1,361**	**98.20**

Conclusion

This paper proposes an algorithm for handwritten-style student identification number segmentation on traditional non-optical grid answer sheets in Thailand. It is an improvement of the segmentation done in our previous work [2], which mainly focused on digit recognition. The algorithm is based on structural analysis, especially profile features. Under the assumption of a digit size correlation in a person's writing, width-height ratio analysis was considered to drive cursive digit segmentation.

Furthermore, profile slope analysis could solve digit pair segmentation well in difficult cases. The number of digits in a sequence must be known a priori. This is reasonable because an SIDN contains a fixed number of digits. This pre-determination facilitates the number of segmentations in each sequence. However, it may also make the error of early-pass segmentation accumulate to the subsequent steps of partitioning since when the desired number of segments is reached, the algorithm ends. The experimentation on 231 SIDNs gave an average accuracy of 98.20%.

The segmentation algorithm along the vertical axis makes some imperfect segments, particularly when the digits are overlapped. Finding segmentation points or segmentation using a slanted vertical line can be developed in future work in order to deal with connections of slanted digits. Moreover, segmentation without reconstruction cannot solve overlapping digits. Therefore, segmentation utilizing reconstruction should be studied to handle these cases.

Acknowledgements

I would like to thank Dr. Parichat Sermwuthisarn for commenting on the writing of my paper, Dr. Desmond Lobo and Dr. Philip Barber for their assistance in correcting my English and the Faculty of Engineering at Kamphaengsaen, Kasetsart University for all the facilities that let me accomplish the work.

References

[1] Sattayakawee N. Test scoring for non-optical grid answer sheet based on projection profile method. Int J Inf Educ Technol 2013;3(2):273-77.

[2] Sattayakawee N, Ratchapaiboon T, Jetpipattanapong D. Handwritten digits OCR for identifying examinee number on objective test answer sheet. NU J Sci Tech 2014;22(2):94-105.

[3] Choi S, Oh I. A segmentation-free recognition of two touching numerals using neural network. Proceedings of the Fifth International Conference on Document Analysis and Recognition; 1999; Bangalore, India. 1999, p.253-6.

[4] Ribas FC, Oliveira LS, Britto Jr, AS, Sabourin R. Handwritten digit segmentation: a comparative study. Int J Doc Anal Recog 2013;16(2):127-37.

[5] Lacerda EB, Mello CAB. Segmentation of connected handwritten digits using Self-Organizing Maps. Expert Syst Appl 2013;40(15):5867-77.

[6] Mello CAB, Roe E, Lacerda EB. Segmentation of overlapping cursive handwritten digits. Proceedings of the eighth ACM symposium on document engineering; 2008; Sao Paulo, Brazil. p. 271-4.

[7] Lopes Filho ANG, Mello CAB. Segmentation of overlapping digits through the emulation of a hypothetical ball and physical forces. Proceedings of the 2015 ACM symposium on document engineering; 2015; Lausanne, Switzerland. p. 223-6.

[8] Wang X, Govindraju V, Srihari S. Holistic recognition of touching digits, Advances in handwriting recognition. World Scientific Publications: Singapore; 1999. p. 359-69

[9] Elnagar A, Alhajj R. Segmentation of connected handwritten numeral strings. Pattern Recogn 2003;36(3): 625-34.

[10] Yu D, Yan H. Reconstruction of broken handwritten digits based on structural morphological features. Pattern Recogn 2001;34(2):235-54.

[11] Ciresan D. Avoiding segmentation in multi-digit numeral string recognition by combining single and two-digit classifiers trained without negative examples. Proceedings of the 2008 10th international symposium on symbolic and numeric algorithms for scientific computing; 2008; Timisoara, Romania. p.225-30.

[12] Rao NV, Babu BR. Segmentation and recognition of handwritten digit numeral string using a multi layer perceptron neural networks. IJFCST 2016;6(1):49-55.

[13] Soares de Oliveira LE, Lethelier E, Bortolozzi F, Sabourin R. Handwritten digits segmentation based on structural approach. Proceedings of XIII Brazilian symposium on computer graphics and image processing; 2000; Gramado, Brazil. p.67-73.

[14] Gattal A, Chibani Y. Segmentation and recognition strategy of handwritten connected digits based on the oriented sliding window. Proceedings of the 2012 international conference on frontiers in handwriting recognition; 2012; Bari, Italy. p. 297-301.

[15] Dimauro G, Impedovo S, Pirlo G, Salzo A. Automatic bankcheck processing: A new engineered system. Int J Pattern Recogn 1997;11(4):467-504.

[16] Suwa M, Naoi S. Segmentation of handwritten numerals by graph representation. Proceedings of 9th international workshop on frontiers of handwriting recognition; 2004; Tokyo, Japan. p. 334-9.

[17] Pal U, Belaïd A, Choisy Ch. Touching numeral segmentation using water reservoir concept. Pattern Recogn Lett 2003;24:261-72.

[18] Otsu N. A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics 1979;9(1):62-6.