Vol. 43 No. 6 IPSJ Journal June 2002

Regular Paper

Detection and Reconstruction Scheme for Broken Characters in Thai Character Recognition Systems

Wichian Premchaiswadi,[†1] Nucharee Premchaiswadi,[†2]
Pongsuree Limmaneewichid[†3] and Seinosuke Narita[†4]

This paper proposes an effective scheme for detection and reconstruction of broken printed Thai character images, which are one of the major problems in character recognition systems. This problem is still unsolved in commercially available Thai character recognition software packages. The proposed reconstruction method is based on the segment boundaries and distinctive features of Thai characters. This scheme consists of two steps: determination of the number of segments belonging to a character and reconstruction of that particular character image. In the first step, it is determined whether each segment is a part of a broken character by using an area of overlap between broken segment boundaries. Then, the number of segments belonging to a character image is determined. The reconstruction techniques are based on the number of broken segments. The projection profile, an overlapping area, and its location are employed to indicate the appropriate areas for connecting broken segments. The test results reveal that the construction is very accurate. The reconstructed characters were also tested by using commercially available Thai character recognition software packages. The test results showed that the proposed scheme significantly improves the recognition rate of the software.

1. Introduction

The accuracy of a character recognition system depends on the completeness of the input character images. Therefore, much research on character recognition systems has focused on pre-processing processes such as segmentation of touching characters[1)], document image binarization[2)], and editing of the curves of characters[3)]. In character recognition systems, broken character images are one major problem that has not been fully explored, especially in Thai OCR. Although much research has been done on Thai character recognition[4),5)], this problem still exists and causes erroneous results in recognition systems. **Table 1** shows an example of a recognition result obtained by using commercially available Thai language OCR software for recognition of broken character images.

Some characters may not be displayable. It is clear that the software cannot recognize most of the broken characters. The broken character problem can occur on account of many causes, such as poor quality of the printing and scanning processes. When a character image is broken into several segments, these segments cannot be treated as a single complete character. A segment is defined as a rectangular area that covers connected pixels. A simple merging of these small segments is not feasible, since it is not clear beforehand which segments belong to which character. Thus, a more advanced technique is required to solve this type of problem: grouping segments of characters to form an entity representing a character. A method for reconstructing damaged characters was proposed in Ref. 6), but it can only be used for characters damaged by lines of a table. It does not address the problem of broken characters and cannot be applied in the case of Thai characters.

This paper proposes a new method based on the connected pixel boundaries and distinctive features of Thai characters for detection and reconstruction of broken printed Thai character images. The method consists of two steps:

(1) determination of the number of segments belonging to a character, and
(2) reconstruction of the character image.

2. Thai Character Set and Multi-level Structure

The Thai character set consists of 44 conso-

[†1] Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand
[†2] Faculty of Information Technology, Dhurakitbundit University, Bangkok, Thailand
[†3] Faculty of Information Science, Mahanakorn Institute of Technology, Bangkok, Thailand
[†4] Department of Electrical, Electronics and Computer Engineering, School of Science & Engineering, Waseda University

nants, 15 vowels, and 8 voice tones, as shown in **Table 2**. The character "อ" in vowels and voice tones is used only to represent the location of vowels and the tone of voice characters. In writing Thai words, it is replaced by one of the consonant characters. A Thai sentence or word is composed of up to three typographical zones: the upper zone, central zone, and lower zone. These zones are defined by four horizontal lines, as shown in **Fig. 1**. A word in a written Thai sentence is formed from a combination of consonants, vowels, and voice tones from different zones. The consonants are located on the baseline, while vowels are located either above the upper line or below the baseline. Voice tones are also located above the upper line. Fortunately, the height of consonant characters written in the central zone is much greater than that of characters in other zones. The height of a consonant character can also be used as one of the criteria for identifying characters in other zones.

Table 1 Example of an OCR recognition result for broken characters.

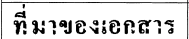
Test data	Software		
	Correct result	ThaiOCR	ArnThai
	ที่มาของเอกสาร	ที่I 16 6 16 'I	ฑูมา.ดขเจจ.ยกสา

Table 2 Thai character set.

Type	Member
Consonants	ก ข ค ง จ ฉ ช ซ ฌ ญ ฎ ฏ ฐ ฑ ฒ ณ ด ต ถ ท ธ น บ ป ผ ฝ พ ฟ ภ ม ย ร ล ว ศ ษ ส ห ฬ อ ฮ
Vowels	ะ า อิ อี อึ อื อุ อู เ แ โ ไ ใ อั อ็
Voice tones	อ่ อ้ อ๊ อ๋ อํ อ์ ฯ ๆ

The multi-level structure of a text line is determined by using the position of the TopLine (to), UpperLine (up), BaseLine (ba), and BottomLine (bo) of each line. The process can be accomplished by using a horizontal projection profile. The areas between "to" and "up", "up" and "ba", and "ba" and "bo", are referred to as the upper zone (UZ), central zone (CZ), and lower zone (LZ) respectively. The upper zone boundary (UZB) is defined as a rectangular area between the points (Xmin, to) and (Xmax, up-1). The central zone boundary (CZB) and lower zone boundary (LZB) are defined as a rectangular area between points (Xmin, up) and (Xmax, ba), and points (Xmin, ba+1) and (Xmax, bo), respectively, where Xmin and Xmax are the minimum and maximum on the x-axis of the document image. Character height (CH) is defined as the distance

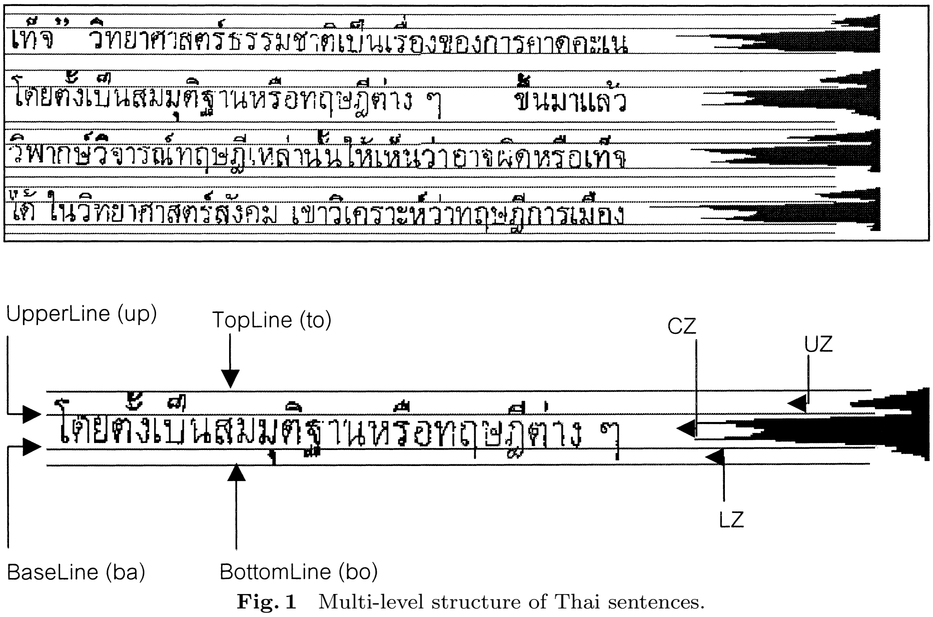


Fig. 1 Multi-level structure of Thai sentences.

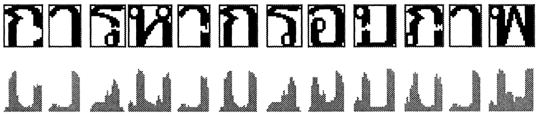


Fig. 2 Result of block segmentation.

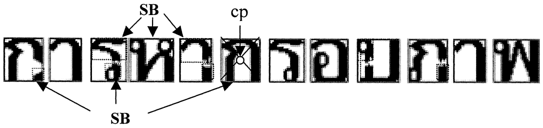


Fig. 3 Result of frame segmentation.

between the BaseLine (ba) and UpperLine (up) (CH = ba − up).

3. Pre-processing

The objective of the pre-processing process is to determine the segment boundaries (SBs) and classify each SB into one of the three zones mentioned in the previous section. An SB is defined as the smallest rectangular area that covers all of a set of connected pixels. Each SB may consist of a single complete character, several connected characters, or a part of a broken character. The pre-processing process consists of two steps: segment boundary determination and zone classification.

3.1 Segment Boundary Determination

The algorithm used for finding each SB consists of two steps: block segmentation and frame segmentation. The block segmentation process is used to separate characters in each line into blocks. This process is simply performed by detecting vertical gaps between characters. Each block may consist of a complete character or broken characters, as shown in **Fig. 2**.

The frame segmentation processes the output of the block segmentation process. In this process, an edge detection algorithm[7] is employed to extract each group of connected pixels from the output of the block segmentation process. The connected pixels and their boundary (SB) can be obtained. The result of the process is shown in **Fig. 3**.

3.2 Zone Classification

This process is used to classify each SB into a specific zone. This zone classification is very useful for determining characteristics and reconstruction methods for each character in the SB. The central point (cp), which is the center of each SB, is used in this classification process. The zone classification checks for the location of the center of each SB and compares cp with the boundary area of each zone in that line. An SB belongs to a specific zone if its center (as indicated by the arrow marked cp in Fig. 3) is within the boundary area of that particular zone.

$$\begin{array}{ll} \mathrm{SB(i)} \in \text{upper zone} & \text{if cp of SB(i)} \in \mathrm{UZB} \\ \mathrm{SB(i)} \in \text{central zone} & \text{if cp of SB(i)} \in \mathrm{CZB} \\ \mathrm{SB(i)} \in \text{lower zone} & \text{if cp of SB(i)} \in \mathrm{LZB} \end{array}$$

Table 3 Zone classification.

Central zone	Lower zone	Upper zone
กขคฆงจฉชซฌญ ฎฏฐฑฒณดตท ธนบปผฝพฟภมฤฦ ยรลวศษสหฬอฮ ะ า เ แ โ ไ ใ ฯ ๆ	อุ อู	อิ อี อึ อื อ่ อ์ อ้ อั อ็ อ๊ อ๋ อํ

By means of zone classification, characters are classified into three zones, as shown in **Table 3**. The characteristics of characters in each zone can be found much more easily. Then, the sequence of each group of SBs is rearranged according to Thai grammar. In Thai grammar, a character in the central zone must be written before a character in the upper zone or lower zone if the characters have the same value of cp on the x-axis. The information of zone classification and the position of the center point of each SB (x-axis) are used for this rearrangement. The output of this process is the sequence of SBs, as shown in **Fig. 4**.

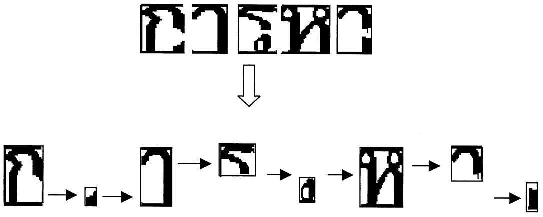


Fig. 4 Sequence of SBs.

4. Analysis of Broken Printed Thai Characters

The broken character problem can occur as a result of many causes such as the scanning process or light print. Light print may be due to weak-fingered typists, misadjusted impact printers, worn ribbons, exhausted printer or copier cartridges, or an inadequate scanning threshold[8]. In the case of a printer with an exhausted toner cartridge, the intensity of the printout may be inconsistent or even seen as a broken line. As a consequence, if that particular printout has been scanned, the quality of the scanned image can suffer as a result of the poor quality of the original.

Documents or forms are typically prepared only as originals (without duplicates). If additional copies are needed, they are made from the original. Each time copies are made, the quality of the image deteriorates, which can result in broken characters. The output quality from a typical facsimile is usually low in terms of clarity, because of low image resolution. In Thailand, broken characters are usually seen in fax-transmitted documents, especially from the transmission of copied (non-original) documents. When rather thin sheets of paper (e.g., 70 gsm) are fed into a printer, small folds are sometimes found, which can result in broken characters in the folded areas.

Sheets which have already been used on one side are often re-used on the other side for economic reasons, and folding can occur easily due to changed characteristics caused by heat from the first use. This phenomenon can often be found in sheets re-used in laser printer or copy machines.

From the reasons stated above, it can be seen that most "broken lines" are in the "legs" of characters, because a "leg" of a character is the longest segment. Furthermore, a "leg" of a character is a narrow segment, and scanning (in a copy machine or facsimile) is performed horizontally, which usually causes the break to occur on the horizontal axis. Depending on the position at which a break occurs, the broken character can be considered as either a vertical segment or a horizontal segment. However, most breaks cause a character to be horizontally broken characters.

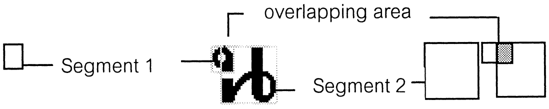


Fig. 5 Segments with overlapping boundaries.

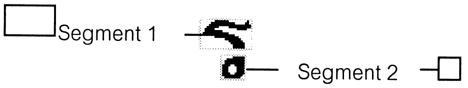


Fig. 6 Segments with non-overlapping boundaries.

Figure 4 shows that a segment can be referred to as either a complete character image or as a part of a character image of a broken character. The boundaries of these segments will be used as criteria for determining a broken character. The distinctive features of Thai characters, such as legs and heads, are also employed to determine the characteristics of characters and to reconstruct characters. For explanatory purposes, the number of legs of a character is defined as the number of vertical lines in that particular character. For example, the character in **Fig. 5** is considered as having two legs, while the characters in **Figs. 6** and **7** (d) are considered as having one and three legs, respectively. A head of a character is defined as a circle or a hole in a character image. Almost all Thai characters have a head, which is used as a starting point in writing that character.

The analysis of broken characters is based on the assumptions that "breaks" in characters occur horizontally and that characters are broken into segments. The types of broken characters can be divided into two major categories according to their broken segment boundaries, as follows:

(1) segmented characters with overlaps be-

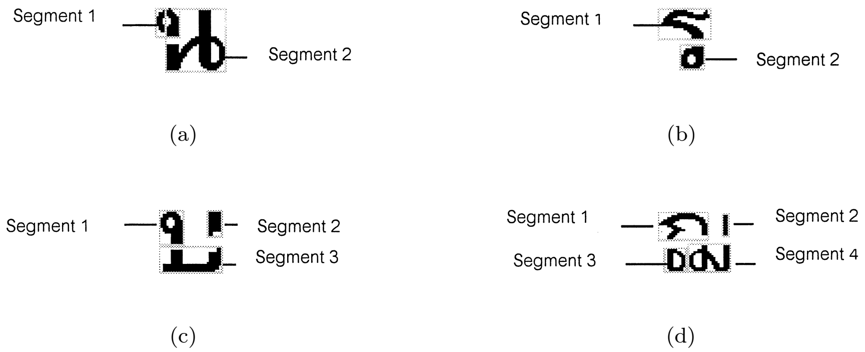
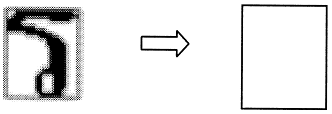
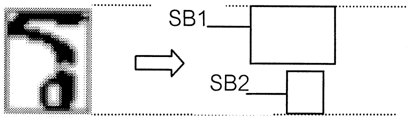


Fig. 7 Many forms of broken characters.



(a) Complete character and its boundary



(b) Broken character and its boundaries

Fig. 8 Broken segment boundaries without overlaps.

tween segment boundaries, as shown in Fig. 5, and

(2) segmented characters without overlaps between segment boundaries, as shown in Fig. 6.

However, broken characters in these two categories can occur in several forms, depending on the shape of the character, as shown in Fig. 7. A different number of broken segments for a character can be obtained from different types of broken characters. The number of broken segments in a character is one of the main criteria for specifying the method for character reconstruction.

In this paper, the analysis of broken characters is limited to a maximum of four segments for any single character, as follows:

4.1 Characters Broken into Two Segments

When a character is broken into two segments, there are two possibilities as regards the boundaries of these two segments: segmented character without overlaps between segment boundaries and segmented character with overlaps between segment boundaries.

4.1.1 Segmented Character without Overlaps between Segment Boundaries

This case occurs only with characters that have a single leg, as shown in **Fig. 8**.

4.1.2 Segmented Character with Overlaps between Segment Boundaries

This case occurs with characters that have at least two legs. The break may occur in the first, second, or third leg of a character, as shown in the example in **Fig. 9**.

Overlaps may occur in different positions and have different sizes, as shown in **Fig. 10**.

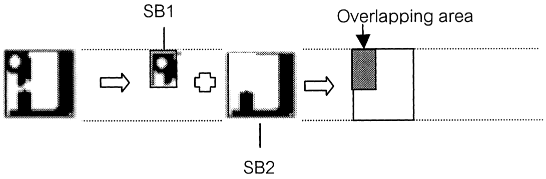
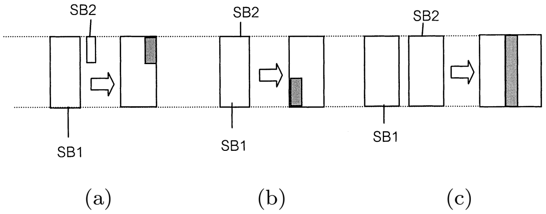
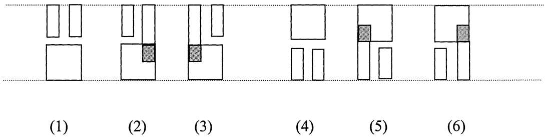


Fig. 9 Example of a broken character with overlap between segment boundaries.



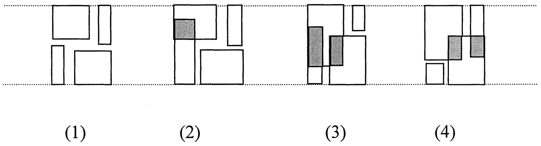
(a) (b) (c)

Fig. 10 Different positions of overlaps.



(1) (2) (3) (4) (5) (6)

Fig. 11 Several cases of characters broken into three segments.



(1) (2) (3) (4)

Fig. 12 Several cases of characters broken into four segments.

4.2 Characters Broken into Three Segments

In this case, the break can occur with characters having at least two legs. All broken segment boundaries may or may not have overlapping areas. Overlapping areas can also occur in many locations, as shown in **Fig. 11**.

4.3 Characters Broken into Four Segments

This case occurs only with characters that have more than two legs. There may or may not be overlaps between these broken boundaries and, if they exist, these overlaps may occur in different positions, as shown in **Fig. 12**.

5. The Proposed Reconstruction Method

The proposed reconstruction method consists of two main steps:

(1) determination of the number of segments belonging to a character, and

(2) reconstruction of the character image.

5.1 Determination of the Number of Segments Belonging to a Character

This process is used to identify a broken character and the number of broken segments that belong to the same character. Non-broken characters will have only one segment for each character. It can be seen from the segmentation process that a segment boundary of a non-broken character will have no overlapping areas with other characters' segment boundaries. As mentioned earlier, there are two types of broken characters: segmented characters with overlaps between the segment boundaries and segmented characters without overlaps between the segment boundaries. In the former case, a segment can be easily recognized as a broken character, because there will be overlaps between segment boundaries. However, the number of segments belonging to that particular character must be determined. In the latter case, the segment may possibly be either a part of a broken character or a complete single character. Therefore, the size and location of each segment must also be taken into account by extending its boundary to the height of that line. If the extended boundary has an overlap with other segments' boundaries, all of the segments will be considered as parts of a broken character, as shown in **Fig. 13**.

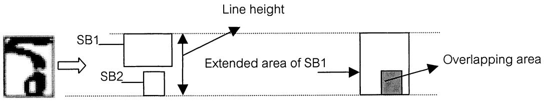


Fig. 13 Overlapping area created by extending the boundary to the top and bottom of the line.

In order to determine the number of segments that belong to the same character, the overlaps between segment boundaries are employed. When a character is broken into several segments, there will be overlaps between these segments if the boundary of each segment is extended to the top and bottom of the line in which it occurs. Then, the number of segments that belong to the same character image and their positions can also be determined. The algorithm used for this process is shown in **Fig. 14**.

5.2 Reconstruction of a Character Image

This section describes a method for reconstructing a character image from the number of broken segments found in the previous section. In this study, we assume that the breaks always occur in the legs of a character. A leg of a character is characterized as a vertical column. Therefore, the reconstruction process consists in filling the gaps in the broken leg with black pixels. According to the number of broken segments, the reconstruction process is divided into the following three categories:

(1) reconstruction from two segments,

(2) reconstruction from three segments, and

(3) reconstruction from four segments.

5.2.1 Reconstruction from Two Segments

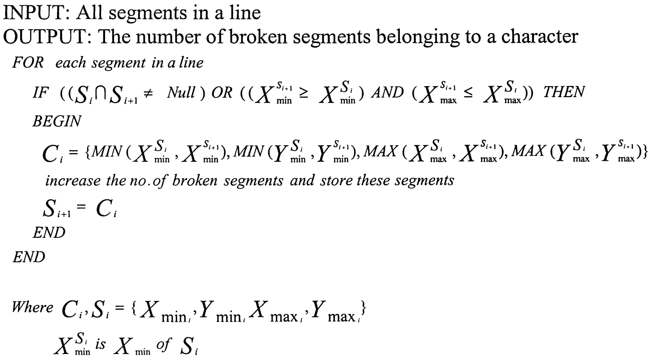
In this case, the process can be divided into the following sub-categories:

(a) reconstruction from segments with an overlap, and

(b) reconstruction from segments without overlaps.

a) Reconstruction from Segments with an Overlap

First, the connection location must be identified. The overlapping area, the boundary of each broken segment, and its position are used



```
INPUT: All segments in a line
OUTPUT: The number of broken segments belonging to a character
 FOR each segment in a line
   IF $((S_i \cap S_{i+1} \neq Null)$ OR $((X_{\min}^{S_{i+1}} \geq X_{\min}^{S_i})$ AND $(X_{\max}^{S_{i+1}} \leq X_{\max}^{S_i}))$ THEN
   BEGIN
     $C_i = \{MIN(X_{\min}^{S_i}, X_{\min}^{S_{i+1}}), MIN(Y_{\min}^{S_i}, Y_{\min}^{S_{i+1}}), MAX(X_{\max}^{S_i}, X_{\max}^{S_{i+1}}), MAX(Y_{\max}^{S_i}, Y_{\max}^{S_{i+1}})\}$
     increase the no.of broken segments and store these segments
     $S_{i+1} = C_i$
   END
 END

 Where $C_i, S_i = \{X_{\min_i}, Y_{\min_i} X_{\max_i}, Y_{\max_i}\}$
     $X_{\min}^{S_i}$ is $X_{\min}$ of $S_i$
```

Fig. 14 Algorithm for determining the number of broken segments in a character.

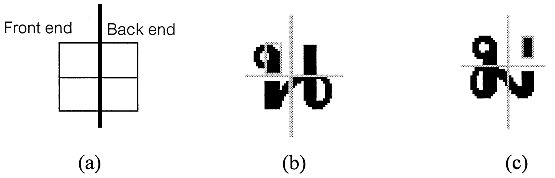


Fig. 15 (a) Four-quadrant separation, (b) the overlapping area in the upper left of the front end, and (c) the overlapping area in the upper right of the back end.

to specify connecting positions. In the Thai character set, there are some distinctive features that affect the reconstruction of characters, such as the location of the head. Therefore, these characteristics must be taken account of in the reconstruction method. The knowledge of the location of the break will be useful in selecting a suitable method for reconstructing a character from the broken segments. The position of the break will be in one of the four quadrants when a character boundary is divided into four quadrants through its center, as shown in **Fig. 15**.

In the case of broken characters, the boundary of each segment may or may not be the boundary of the complete character. Therefore, the boundary of the complete character must be determined. This can be obtained by finding the area that can cover all these broken segments. The algorithm for finding the complete boundary of a character is shown in Eq. (1).

$$C\_Boundary = \left\{ \underset{i=1\ to\ n}{\mathrm{MIN}}\left(X^{S_i}_{\min}\right), \underset{i=1\ to\ n}{\mathrm{MIN}}\left(Y^{S_i}_{\min}\right), \underset{i=1\ to\ n}{\mathrm{MAX}}\left(X^{S_i}_{\max}\right), \underset{i=1\ to\ n}{\mathrm{MAX}}\left(Y^{S_i}_{\max}\right) \right\} \quad (1)$$

where $n = 2$ *to* 4, *depending on the number of broken segments.*

The location of the overlap can be classified into one of the four quadrants mentioned above by checking the location of the center of the overlapping area. The reconstruction method is designed on the basis of the location of the overlap and is divided into two further categories, namely:

1.1) reconstruction from the back end (upper right and lower right), and

1.2) reconstruction from the front end (upper left and lower left).

To explain the reconstruction method, a segment boundary is represented by the upper left corner $(x1, y1)$ and lower right corner $(x2, y2)$, and the overlapping area is defined by $(Overlap.X_{\min}, Overlap.Y_{\min})$ and $(Overlap.X_{\max}, Overlap.Y_{\max})$, as shown in **Fig. 16**.

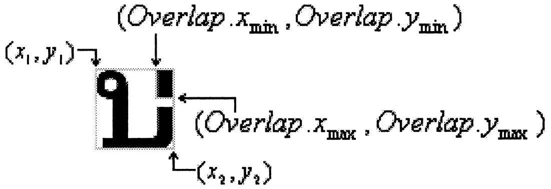

$(Overlap.x_{\min}, Overlap.y_{\min})$
(x_1, y_1)
$(Overlap.x_{\max}, Overlap.y_{\max})$
(x_2, y_2)

Fig. 16 Representation of the segment boundaries and overlapping area.

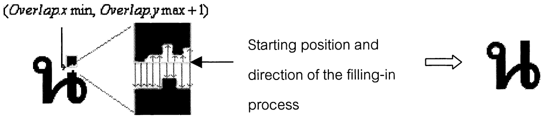

(Overlap.x min, Overlap.y max + 1)
Starting position and direction of the filling-in process

Fig. 17 Reconstruction of the upper back end.

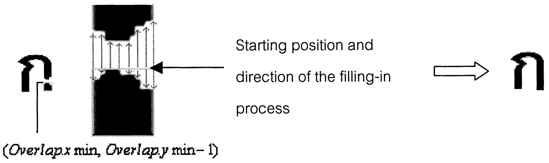

Starting position and direction of the filling-in process
(Overlap.x min, Overlap.y min − 1)

Fig. 18 Reconstruction of the lower back end.

1.1) Reconstruction from the Back End

The reconstruction process is divided into two cases, depending on the locations of the overlapping area:

a) The overlapping area is in the upper-right quadrant: In this case, the width of the overlapping area is used as the width for connection. The reconstruction is performed at the bottom line of the overlapping area boundary and fill-in black pixels are added in both the upward and downward directions from this point until a black pixel or the boundary of the character is reached, as shown in **Fig. 17**.

b) The overlapping area is in the lower-right quadrant: In this case, the starting point for filling in with black pixels is the top line of the overlapping area boundary. The method of filling in with black pixels is the same as the one mentioned above, as shown in **Fig. 18**.

1.2) Reconstruction of the Front End

In this case, the reconstruction of a character involves the head of the character. Improper determination of the connection points will lead

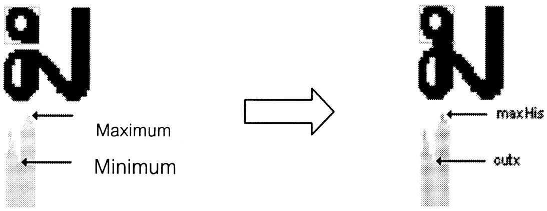


Fig. 19 Projection profile of the front end part.

to an output quite different from the correct one. The result will cause problems in the recognition process. To solve this kind of problem, the projection profile[6] is used to specify the proper points for connecting segments. The projection profile is calculated not only from all overlapping areas but also from the area obtained from Eq. (2).

Pr*ojection_area* =

$$\left\{X^{O}_{\min},\ \mathrm{MIN}\left(Y^{O}_{\min}, Y^{S_i}_{\min}, Y^{S_{i+1}}_{\min}\right),\ X^{O}_{\max},\ \mathrm{MAX}\left(Y^{O}_{\max}, Y^{S_i}_{\max}, Y^{S_{i+1}}_{\max}\right)\right\} \quad (2)$$

where O is the overlapping area of S_i *and* S_{i+1}.

From **Fig. 19**, it can be found that the maximum value in the projection profile is in the leg area of a character. The lowest point can be on either the left-hand side or the right-hand side of the maximum value point, depending on the direction of the head. If the direction of the head is outside the character, as shown in Fig. 19, black pixels will be added from the minimum point of the projection profile to the *Overlap.X*$_{\max}$. If the direction of the head is inside the character, black pixels will be added from the minimum point of the projection profile to the *Overlap.X*$_{\min}$.

b) Reconstruction from Segments without Overlaps

As mentioned earlier, the overlapping area in this case can be found by extending the segment boundary of the larger segment to the top and bottom of the line in which it occurs. Then, the connection position will be found by using the projection profile as shown in **Fig. 20**. The reconstruction process is performed in the same way as described earlier.

5.2.2 Reconstruction from Three Segments

This case occurs with characters that have at least two legs. The consideration will focus on the width of each segment, and can be divided into the following two cases:

a) the widest segment is in the upper half, and

b) the widest segment is in the lower half.

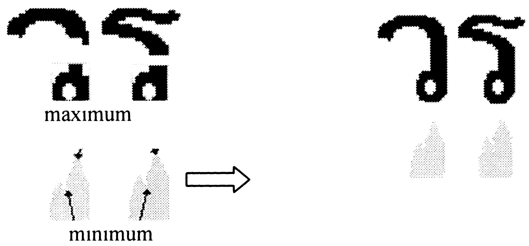


Fig. 20 Reconstruction of characters from segments without overlaps.

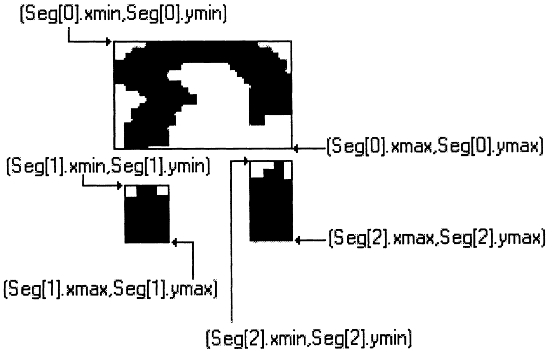

(Seg[0].xmin,Seg[0].ymin)
(Seg[0].xmax,Seg[0].ymax)
(Seg[1].xmin,Seg[1].ymin)
(Seg[2].xmax,Seg[2].ymax)
(Seg[1].xmax,Seg[1].ymax)
(Seg[2].xmin,Seg[2].ymin)

Fig. 21 Example of a character broken into three segments.

The width and location of a segment are defined as follows:

$$\text{Width} = \text{xmax} - \text{xmin} + 1 \quad (3)$$

$$(\text{ymax} + \text{ymin})/2 - (\text{y2} - \text{y1})/2 > 0$$
$$\rightarrow \text{ segment is in upper half}$$
$$(\text{ymax} + \text{ymin})/2 - (\text{y2} - \text{y1})/2 < 0$$
$$\rightarrow \text{ segment is in lower half}$$

a) The widest segment is in the upper half

An example in which the widest broken segment is in the upper half is shown in **Fig. 21**.

From Fig. 21, it can be seen that the widest segment is *Seg*[0], (*Seg*[0].xmax − *Seg*[0].xmin + 1), and is located in the upper half, (*Seg*[0].ymin0 + *Seg*[0].ymax)/2 > (y2 − y1)/2.). The next step is to find the overlap between this segment and other segments by extending the boundary of this segment (*Seg*[0]) by changing *Seg*[0].ymax to y2, as shown in **Fig. 22**.

The connecting process begins with the pair *Seg*[0] and *Seg*[1] and then *Seg*[0] and *Seg*[2]. Both of these cases can be considered as the connection of segments with overlaps, as shown in **Figs. 23** and **24**, respectively. The result of these connections is shown in **Fig. 25**.

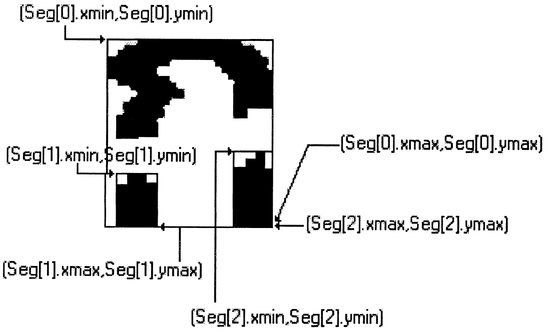

(Seg[0].xmin,Seg[0].ymin)
(Seg[1].xmin,Seg[1].ymin)
(Seg[0].xmax,Seg[0].ymax)
(Seg[2].xmax,Seg[2].ymax)
(Seg[1].xmax,Seg[1].ymax)
(Seg[2].xmin,Seg[2].ymin)

Fig. 22 Changing the coordinate *Seg*[0].ymax to y2.

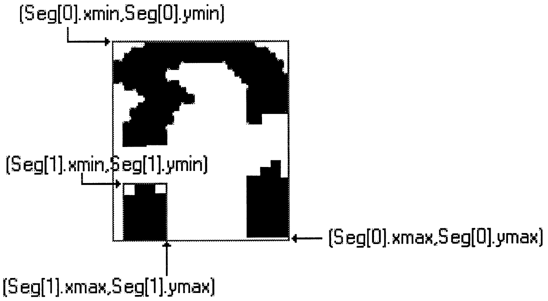

(Seg[0].xmin,Seg[0].ymin)
(Seg[1].xmin,Seg[1].ymin)
(Seg[0].xmax,Seg[0].ymax)
(Seg[1].xmax,Seg[1].ymax)

Fig. 23 Overlap between *Seg*[0] and *Seg*[1].

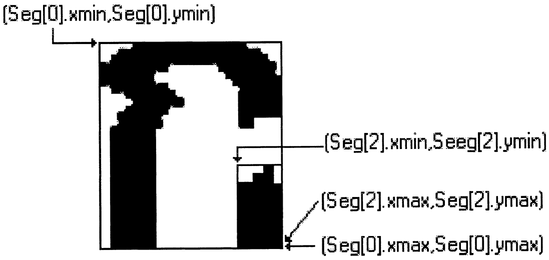

(Seg[0].xmin,Seg[0].ymin)
(Seg[2].xmin,Seeg[2].ymin)
(Seg[2].xmax,Seg[2].ymax)
(Seg[0].xmax,Seg[0].ymax)

Fig. 24 Overlap between *Seg*[0] and *Seg*[2].

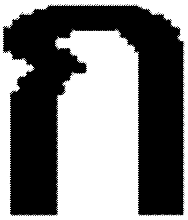


Fig. 25 Result of reconstructing the broken character in Fig. 21.

b) The widest segment is in the lower half

An example in which the widest broken segment is in the lower half is shown in **Fig. 26**.

In this case, the widest segment is *Seg*[2], *Seg*[2].xmax − *Seg*[2].xmin + 1, and it is in the lower half, because ((*Seg*[2].ymin + *Seg*[2].xmax)/2) < (($y2 + y1$)/2). The overlapping area can be found by extending *Seg*[2].ymin to y1, as shown in **Fig. 27**.

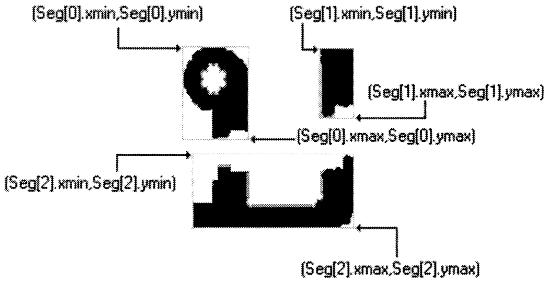

(Seg[0].xmin,Seg[0].ymin)
(Seg[1].xmin,Seg[1].ymin)
(Seg[1].xmax,Seg[1].ymax)
(Seg[0].xmax,Seg[0].ymax)
(Seg[2].xmin,Seg[2].ymin)
(Seg[2].xmax,Seg[2].ymax)

Fig. 26 The widest segment is in the lower half.

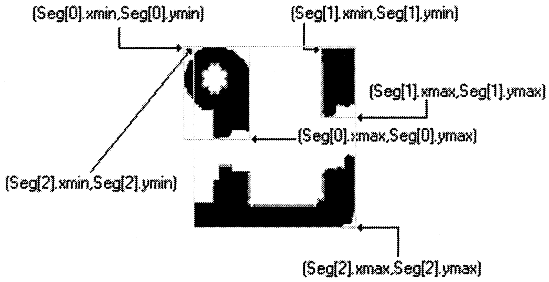

(Seg[0].xmin,Seg[0].ymin)
(Seg[1].xmin,Seg[1].ymin)
(Seg[1].xmax,Seg[1].ymax)
(Seg[0].xmax,Seg[0].ymax)
(Seg[2].xmin,Seg[2].ymin)
(Seg[2].xmax,Seg[2].ymax)

Fig. 27 Extending *Seg*[2].ymin to y1.

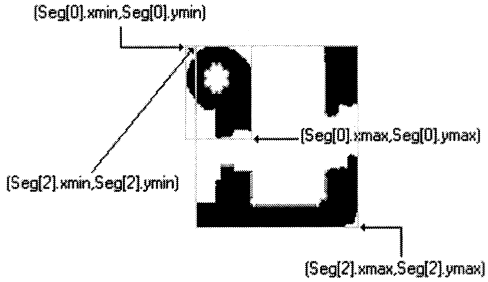

(Seg[0].xmin,Seg[0].ymin)
(Seg[0].xmax,Seg[0].ymax)
(Seg[2].xmin,Seg[2].ymin)
(Seg[2].xmax,Seg[2].ymax)

Fig. 28 Overlaps between *Seg*[2] and *Seg*[0].

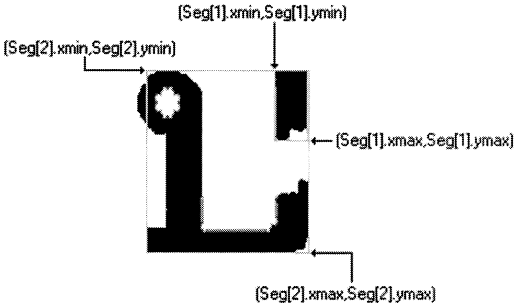

(Seg[1].xmin,Seg[1].ymin)
(Seg[2].xmin,Seg[2].ymin)
(Seg[1].xmax,Seg[1].ymax)
(Seg[2].xmax,Seg[2].ymax)

Fig. 29 Overlaps between *Seg*[2] and *Seg*[1].

Then, processes for connecting *Seg*[2] and *Seg*[0] and *Seg*[2] and *Seg*[1] are performed. The operation is performed in the same fashion as that for segments having an overlapping area, which we described earlier. The results of this process are shown in **Figs. 28** and **29**,

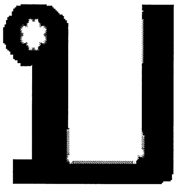


Fig. 30 Result of reconstructing the broken character in Fig. 26.

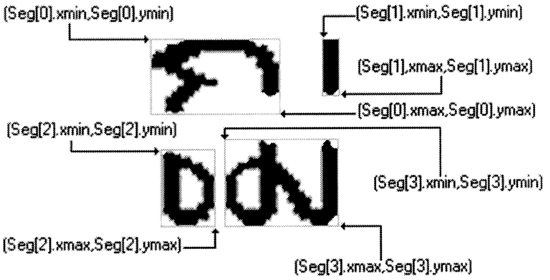

[Seg[0].xmin,Seg[0].ymin]
[Seg[1].xmin,Seg[1].ymin]
[Seg[1].xmax,Seg[1].ymax]
[Seg[0].xmax,Seg[0].ymax]
[Seg[2].xmin,Seg[2].ymin]
[Seg[3].xmin,Seg[3].ymin]
[Seg[2].xmax,Seg[2].ymax]
[Seg[3].xmax,Seg[3].ymax]

Fig. 31 Example of a character broken into four segments.

respectively.

5.2.3 Reconstruction from Four Segments

The coordinates of four broken segments are shown in **Fig. 31**.

First, the sequence of a broken character must be determined for use in the connection process. The sequence is defined by using the center in the x direction, as shown in Eq. (4). The sequence is sorted in ascending order.

$$Xcenter[n] = (Seg[n].\mathrm{xmin} + Seg[n].\mathrm{xmax})/2 \qquad (4)$$

For example, the centers of the segments in Fig. 31 are sorted in the following sequence:

$$X_{center}[2] < X_{center}[0] < X_{center}[3] < X_{center}[1]$$

The sequence for operations on broken segments is $Seg[2]$, $Seg[0]$, $Seg[3]$, and $Seg[1]$, respectively. In this process, $Seg[0]$ and $Seg[3]$ are usually wider than the others. The connecting process is performed for the first two broken segments in the list and then repeated for the next segments in the sequence until the end of the list.

a) Connection of $Seg[0]$ and $Seg[2]$,
$Seg[0]$ is usually wider than $Seg[2]$. The process begins by finding the overlap between these two segments by extending $Seg[0]$.ymax to y2. The connecting process is the same as for segments having an overlapping area, as shown in **Fig. 32**.

b) Connection of $Seg[0]$ and $Seg[3]$,

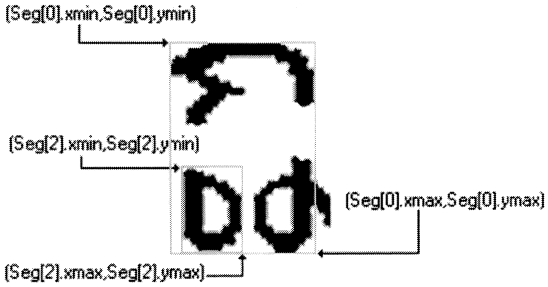

[Seg[0].xmin,Seg[0].ymin]
[Seg[2].xmin,Seg[2].ymin]
[Seg[0].xmax,Seg[0].ymax]
[Seg[2].xmax,Seg[2].ymax]

Fig. 32 Extending $Seg[0]$.ymax to y2.

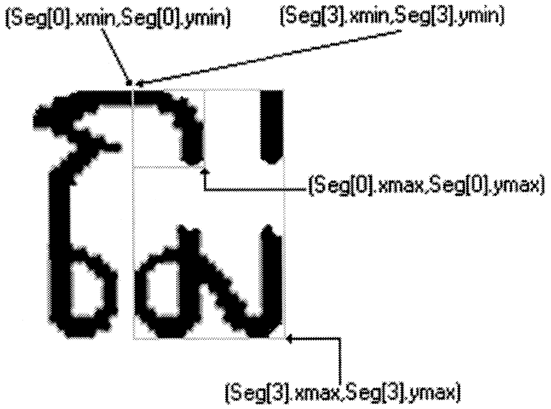

[Seg[0].xmin,Seg[0].ymin]
[Seg[3].xmin,Seg[3].ymin]
[Seg[0].xmax,Seg[0].ymax]
[Seg[3].xmax,Seg[3].ymax]

Fig. 33 Finding an overlap between $Seg[0]$ and $Seg[3]$.

The overlap is found by changing the coordinates as follows:

$Seg[0]$.xmin is replaced by
$Seg[3]$.xmin and
$Seg[3]$.ymin is replaced by
$Seg[0]$.ymin.

The connection area is found as shown in **Fig. 33**. After that, the connecting process is the same as in the case where there is an overlap.

c) Connection of $Seg[1]$ and $Seg[3]$,
The process also begins by finding the overlap. Then, the broken segments are connected in the manner described earlier, as shown in **Fig. 34**. The complete result of this operation is shown in **Fig. 35**.

6. Experimental Results

The method was implemented in Visual C++ Version 6.0. The test documents were obtained from images scanned at a resolution of 600 dpi. These broken characters images have different fonts and sizes. **Figure 36** shows an example of broken character images and their correspond-

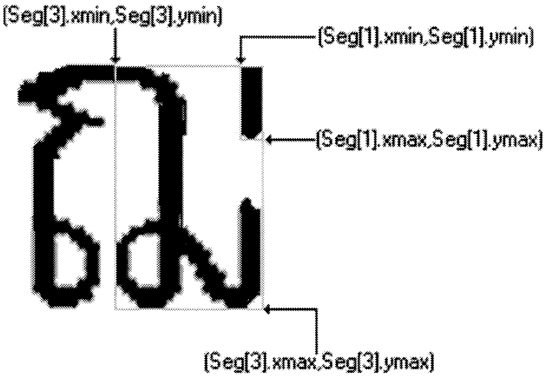

(Seg[3].xmin,Seg[3].ymin)
(Seg[1].xmin,Seg[1].ymin)
(Seg[1].xmax,Seg[1].ymax)
(Seg[3].xmax,Seg[3].ymax)

Fig. 34 Overlap between *Seg*[1] and *Seg*[3].

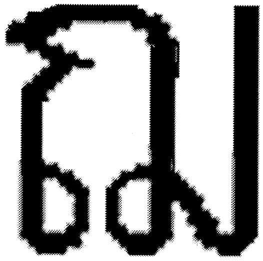


Fig. 35 Final result of the connection.

(a) Broken characters (b) After reconstruction

Fig. 36 Example of a reconstruction result.

ing reconstruction result.

The effectiveness of the proposed scheme was obtained by using commercially available Thai character recognition software. **Figure 37** shows another example of broken character images.

The recognition results for the character images shown in Fig. 37, using the above mentioned commercially available Thai character recognition software, are presented in **Table 4**. These results are also in the same format as the output of the commercial software used in the comparison. It can be clearly seen that the software can hardly recognize any of these broken characters.

The broken character images in Fig. 37 were then reconstructed by the proposed scheme. **Figure 38** shows the results.

The recognition results obtained by using the same software as in Table 4 for the reconstructed images in Fig. 38 are shown in

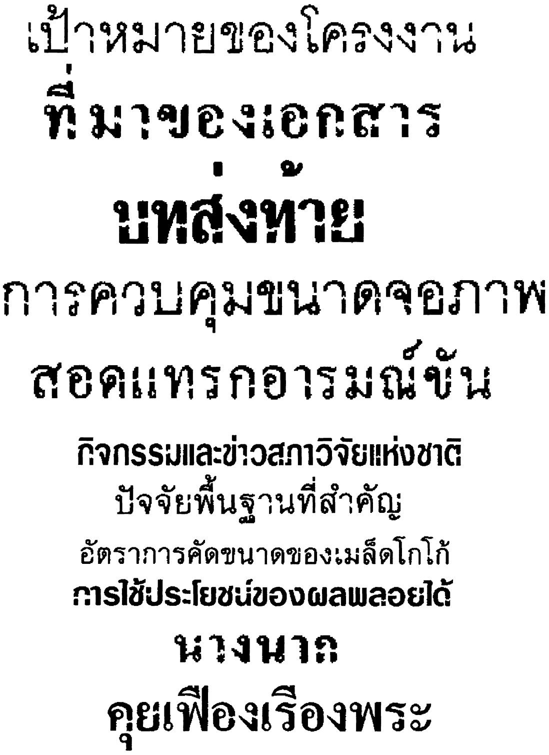


Fig. 37 Examples of broken character images.

Table 4 Recognition results for the broken character images in Fig. 37.

Correct results	Software	
	ThaiOCR	ArnThai
เป้าหมายของโครงงาน	ฐุ 6บฯ'Aมาปี ชิ่ N ค.ฐ๊NNี่ศ่	บจจมายรขๆจจค๑ยจบจจ
ที่มาของเอกสาร	ฟี่▌16 6 16 '▌	ฟู้มา.คขเจจ.ยกสา
บทส่งท้าย	-	พหหยจหจ,.
การควบคุมขนาดจอภาพ	-	กฤๅคกุจ .ค9ขจจคจุผกพฒ ฤ
สอดแทรกอารมณ์ขัน	-	ั จ , .
กิจกรรมและข่าวสภาวิจัยแห่งชาติ	ก่าฤๅ8พต ล.่วสถาว่ายห่จชาณ่	
ปัจจัยพื้นฐานที่สำคัญ	บฯจจัย.พ่บจาจนหี่สำขคัญ	
อัตราการคัดขนาดของเมล็ดโกโก้	ัยตจาก ารคัดขนาดขอฝจมลี็ดโกโขก	ยิ้ตจ.าก. ารคัดขนาดข่อ□ุbม่ลี็ดโกโbก้
การใช้ประโยชน์ของผลพลอยได้	ใ8ใช้ป 8ขยช.ข้องผลพลอย.ไฉ้	s□□MJ□ อkะฐ □◢ □00Ma□ล.□□□0ะ
นางนาก	น.,ขงนาณ.	น่▌ง□▌.
คุยเฟื่องเรื่องพระ	คฒย.เฟี่ฒย๒จเรื่องพโ๓ะ	คุ.ย'เฟี่16Nี่เรื่อง▌ะ

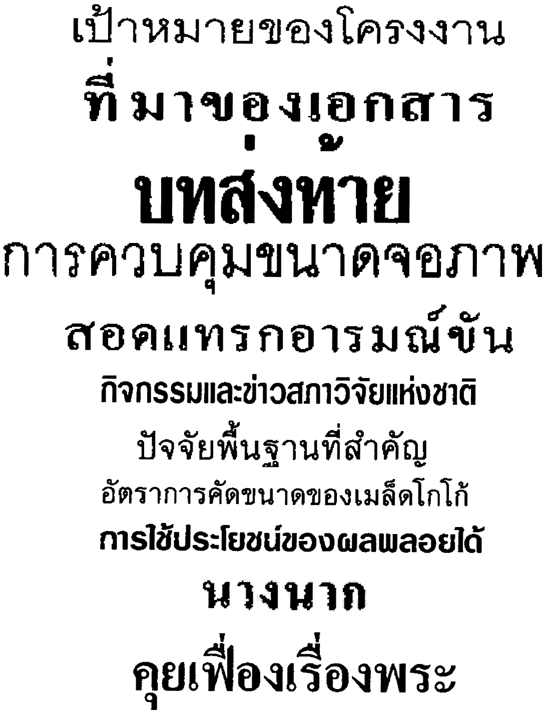


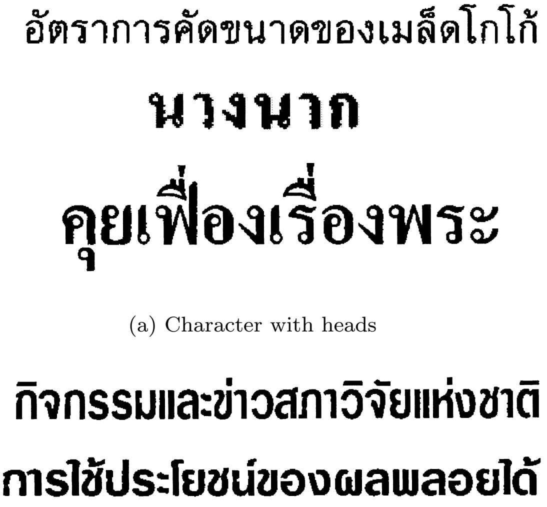
Fig. 38 Character images after reconstruction from Fig. 37.

Table 5 Recognition results after reconstruction of the broken characters in Fig. 39.

Correct Result	Software	
	ThaiOCR	ArnThai
เป้าหมายของโครงงาน	เ าหมายของโครงงาน	เปาหมายของโครงงาน
ที่มาของเอกสาร	ที่ มาของเอก าร	ที่มาของเอกสาร
บทส่งท้าย	ส่ง้าย	บหสงหาย
การควบคุมขนาดจอภาพ	การควบคุมขนาดจอภาพ	การควบคุมขนาดจอภาพ
สอดแทรกอารมณ์ขัน	สอดเ ทรกอารมณ์ขัน	สอดเทรกอารมณ์ขัน
กิจกรรมและข่าวสภาวิจัยแห่งชาติ	ก่าฤๅ8ฤๅลข.่าวสกาว่ายห่จชา๓่	
ปัจจัยพื้นฐานที่สำคัญ	ปจจัยพ่นใ.านที่สำคัญ	ปจจัยพื้นฐานทีสำคัญ
อัตราการคัดขนาดของเมล็ดโกโก้	อัตราการคัดขนาดของเมล็ดโกโก้	อัตราการคัดขนาดของเมล็ดโกโก้
การใช้ประโยชน์ของผลพลอยได้	ถใ8ใช้ป8ขยชน์องผลพลอยไ๋Ծ	
นางนาก	นางนาก	นาง นา
คุยเฟื่องเรื่องพระ	คุยเฟื่องเรื่องพระ	คุยเฟื่องเรื่องพระ

Table 5.

The proposed scheme was tested with various fonts and character sizes, as shown in Fig. 37. The reconstructed character images obtained by using the proposed scheme are shown in Fig. 38. The recognition results obtained by using two Thai OCR packages for both broken and reconstructed characters are shown in Tables 4 and 5. It can be seen from the results that the recognition rate depended on the algorithm and performance of each software package. Both the software packages gave good recognition results for characters with font types that have heads, as shown in **Fig. 39** (a), while neither could recognize characters with font types that do not have heads, as shown in Fig. 39 (b). For Thai people, the reconstructed character images in Fig. 39 (b) are visually considered as complete characters. Therefore, the recognition rate may not be used as the only indicator for the success of the proposed scheme. However, the overall recognition rate for the two software packages is significantly improved by the used of the proposed scheme, as shown in **Table 6**.


อัตราการคัดขนาดของเมล็ดโกโก้
นางนาก
คุยเฟื่องเรื่องพระ
กิจกรรมและข่าวสภาวิจัยแห่งชาติ
การใช้ประโยชน์ของผลพลอยได้

(a) Character with heads

(b) Character without heads

Fig. 39 Characters with and without heads.

From Table 6, it can be seen that the recognition rate is improved significantly, especially for characters with font types that have heads. The overall recognition rate for the ThaiOCR software improved by approximately 48 percent (from 19% to 67%), while the overall recognition rate for the ArnThai software improved by approximately 43 percent (from 32% to 75%).

The proposed scheme cannot be applied to characters broken along the vertical line, because of the assumptions that

a) the break occurs in the "leg" of a character, and

b) an overlap can be found on the x-axis.

Table 6 Comparison of recognition results.

Case	Test text (character count)	ThaiOCR Correct results (No. of characters) Before	ThaiOCR Correct results (No. of characters) After	ArnThai Correct results (No. of characters) Before	ArnThai Correct results (No. of characters) After
1	เป้าหมายของโครงงาน (18)	3 (16.67%)	16 (88.89%)	3 (16.67%)	17 (94.44%)
2	ที่มาของเอกสาร (14)	0 (0%)	13 (92.86%)	5 (35.71%)	14 (100%)
3	บทส่งท้าย (9)	0 (0%)	5 (55.56%)	0 (0%)	5 (55.56%)
4	การควบคุมขนาดจอภาพ (18)	0 (0%)	18 (100%)	1 (5.56%)	18 (100%)
5	สอดแทรกอารมณ์ขัน (16)	0 (0%)	15 (93.75%)	0 (0%)	15 (93.75)
6	กิจกรรมและข่าวสภาวิจัยแห่งชาติ (30)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
7	**ปัจจัยพื้นฐานที่สำคัญ (22)**	0 (0%)	20 (90.90%)	10 (45.45%)	18 (81.81%)
8	อัตราการคัดขนาดของเมล็ดโกโก้ (28)	21 (75.0%)	28 (100%)	20 (71.42%)	28 (100%)
9	การใช้ประโยชน์ของผลพลอยได้ (26)	0 (0%)	0 (0%)	15 (57.69%)	15 (57.69%)
10	นางนาก (6)	2 (33.33)	5 (83.33%)	2 (33.33%)	6 (100%)
11	คุยเฟื่องเรื่องพระ (18)	13 (72.22%)	18 (100%)	10 (55.56%)	18 (100%)
	Average recognition rate	19.02%	67.31%	32.19%	75.12%

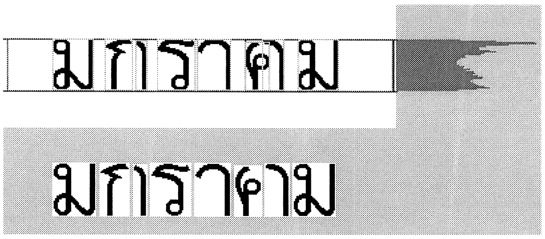


Fig. 40 Vertically broken characters.

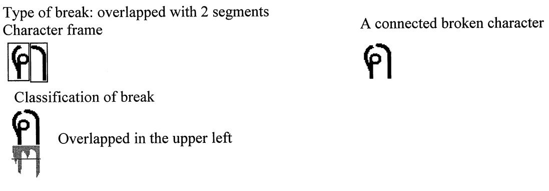

Type of break: overlapped with 2 segments
Character frame
A connected broken character
Classification of break
Overlapped in the upper left

Fig. 41 Detection of broken characters.

However, the proposed scheme can detect vertically broken characters if they have overlaps, as shown in **Fig. 40**. In the case of the character "ค" in Fig. 40, it can be seen that there is an overlap between broken segments. Therefore, the proposed scheme can detect the break but cannot connect it, as shown in **Fig. 41**.

In the case of the vertically broken character "ก", the proposed scheme cannot detect whether it is a broken character, because of the lack of information.

Some characters consist of two parts, such as "ะ", "แ" and "ฐ". If the character "ะ" is broken into pieces, the break will not occur in the leg and the character is usually referred to as vertically broken. Therefore, it is not covered in the proposed scheme. For the character "แ", the pre-process will separate the character "แ" into two characters "เ" and operate on them as if each were the character "เ". The character "ฐ" can be considered as having two parts or one part, depending on the font used. In the pre-process, the boundary of each zone is known. The lower part of character "ฐ" is in the lower zone, and is not used to analyze the overlap. The other operations are the same as for other characters.

7. Conclusion

A scheme for detection and reconstruction of broken printed Thai characters has been proposed. The scheme is based on segment boundaries and distinctive features of Thai characters. Segment boundaries can be simply obtained, and are very useful in detecting and determining the number of broken segments that belong to the same character. The use of a projection profile and the overlapping areas of broken segments make the reconstructed character images very similar to the original characters. It can be seen that the correctness of the recognition results in Table 5 is much higher than that of the results shown in Table 4. However, there are still some erroneous recognition results, although the reconstructed characters are very similar to the original characters. The rate of recognition errors also depends on the efficiency of the recognition software itself. From the experimental results, it can be concluded that the proposed scheme can be used to reconstruct broken characters efficiently and will be useful in significantly improving the recognition rate of Thai character recognition systems.

References

1) Premchaiswadi, N., Premchaiswadi, W. and Narita, S.: Segmentation of Horizontal and Vertical Touching Thai Characters, *ITC-CSCC'99 International Technical Conference on Circuit Systems, Computers and Communications*, Nigata, Japan, pp.25–28 (1999).

2) Kamel, M. and Zhao, A.: Extraction of Binary Character/Graphics Images from Grayscale

Document Images, *Computer Vision, Graphics and Image Processing 55*, pp.203–217 (May 1993).

3) Makkun, C., Koosirivanichakorn, P., Chitsakul, K. and Sangworisil, M.: Editing the Character of a Curve by Wavelets, *RESTECS '96 Regional Symposium on Telecommunications, Electronic Circuits, and Systems*, Bangkok, Thailand.

4) Hiranvanichakorn, P. and Boonsuwan, M.: Recognition of Thai Characters, *Proc. SNLP '93*, pp.123–166 (1993).

5) Kimpan, C. and Walairacht, S.: Thai Character Recognition, *Proc. SNLP '93*, pp.196–260 (1993).

6) Lee, K., Byun, H. and Lee, Y.: Robust Reconstruction of Damaged Character Images on the Form Documents, *Lecture Notes in Computer Science 1389*, pp.149–162, Springer Verlag (1998).

7) Davies, E.R.: *Machine Vision*, Academic Press (1997).

8) Rice, S.V., Nagy, G. and Nartker, T.A.: *Optical Character Recognition: An Illustrated Guide to the Frontier*, Kluwer Academic Publishers (1999).

(Received September 25, 2000)
(Accepted March 14, 2002)

Wichian Premchaiswadi received his D. Eng. degree in electrical engineering from Waseda University, Japan in 1992. He is currently the Dean and associate professor at the Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang (KMITL). His research interests include character recognition, multimedia, database, parallel processing and data mining.

Nucharee Premchaiswadi received her D. Eng. degree in electrical engineering from Waseda University, Japan in 2001. She is currently the Dean and associate professor at the Faculty of Information Technology, Dhurakitbundit University. Her research interests include character recognition, multimedia and CAI and CAL.

Pongsuree Limmaneewichid received his M.S. degree in information technology from King Mongkut's Institute of Technology Ladkrabang, Thailand, in 2000. He is currently a lecturer in Department of Information Technology and Head of Network Operation Center at the Mahanakorn University of Technology. His research interests include character recognition system, fingerprint recognition, and network and data communication applications.

Seinosuke Narita received his D. Eng. degree in electrical engineering from Waseda University, Japan in 1968. He is currently the professor at Department of Electrical, Electronics and Computer Engineering, Waseda University. His research interests include character recognition, multimedia, database, parallel processing, CAI, CAL and discrete system.