Line Segmentation Approach for Ancient Palm Leaf Manuscripts using Competitive Learning Algorithm

Dona Valy[*†], Michel Verleysen[*], and Kimheng Sok[†]
[*]ICTEAM Institute, Université Catholique de Louvain, Belgium
[†]Department of Information and Communication Engineering, Institute of Technology of Cambodia, Cambodia
{dona.valy,michel.verleysen}@uclouvain.be, sok.kimheng@itc.edu.kh

Abstract— **Line segmentation is very crucial in handwritten text recognition/analysis task. A new text line extraction scheme based on a data clustering algorithm is proposed. Our approach starts by determining the number of lines and setting up text line mid points' initial positions using a modified piece-wise projection profile technique. We apply afterwards competitive learning algorithm to adaptively move those mid points according to the geometrical information of connected components in the document page to form lines. Borders between text lines are defined so that they can be used to separate touching components that spread over multiple lines. The proposed method is robust in handling documents with skewed, fluctuated, or discontinued text lines. Experimental evaluations were made on a data set of Khmer ancient palm leaf manuscripts.**

Keywords-handwritten document analysis; text line segmentation; competitive learning algorithm

I. INTRODUCTION

Dating back as early as the fifth century B.C., palm leaves were largely used as a writing material in many countries especially in South and South-East Asia [1]. In Cambodia, palm leaf documents are still seen in Buddhist establishments (mostly Buddhist temples called "pagoda") and are considered to be holy and sacred. They are being used habitually and traditionally by monks as reading scriptures. Besides their cultural merit, these ancient manuscripts store valuable content including old religious sayings, crucial historical events, as well as certain scientific findings that are still very useful for researchers in those fields of study.

As a consequence of deterioration from natural aging and damage caused by moisture, fungus, and insects, palm leaf manuscripts are facing destruction and are in need for preservation. Many programs and projects are underway to recover and preserve palm leaf documents not only in their physical form but also in digital imaging through scanning and photography. The centralization of the digitized images allows easy access for the public, for instance a collaboration effort with EFEO (Ecole Française d'Extrême-Orient) for Khmer palm leaf documents [2]. Nonetheless, searching and filtering the content of those documents using particular keywords are still unmanageable. An automatic recognition system therefore needs to be developed.

Line separation is one of the most important and challenging pre-processing tasks in handwritten character recognition especially for the case of old Khmer handwriting on palm leaf documents. Each individual character is carved using a sharp writing tool on long rectangular-cut dried palm leaves. The main difficulties in extracting lines from these documents result from the fact that lines are cramped up together with very little spacing between them due to relatively small height of the documents compared to their wide width (Fig. 1). In addition, the characteristic of Khmer alphabet, which consists of a large number of consonants, vowels, and special characters, is also a big challenge attributable to the irregular position of how those characters are combined into words. Unlike the writing of most Latin languages where characters are sequenced from left to right, in Khmer writing, vowels are positioned either on the left, on the right, below, or above the consonant they are spelled with. Two or more consonants can also be merged together by transforming into different shapes called low-consonant or subscript form which are placed below the main consonant as shown as an example in Fig. 2. This variation of how Khmer letters are formed produces multiple levels (sometimes more than three) of characters, and they tend to go far out of their main line, touch, or overlap with other characters from adjacent lines.

Text lines may also be slanted or curved upward/downward on account of it being handwritten or from improper digitizing. Furthermore, holes used for binding the manuscript leaves leave behind empty areas in the middle of the page creating discontinuity of text lines. Fig. 1 illustrates how the three middle lines are separated into sections due to the two holes.

In this paper, we propose a segmentation method focusing on line extraction from ancient Khmer palm leaf manuscripts. The method is based on identifying connected components into correct lines using adaptive clustering algorithm on the geometrical information of the components. Our approach is able to detect fluctuated and discontinued lines; it yields a good result on our testing data set.

The paper is structured as follows: Section II briefly discusses existing literature related to line extraction methods and their efficiency and usefulness for segmenting ancient palm leaf documents. Section III describes in specific details our approach proposed to segment palm leaf manuscripts into lines, some challenges encountered and how to solve them. The experimental results are presented in Section IV. Final conclusions and perspectives for future work are given in Section V.

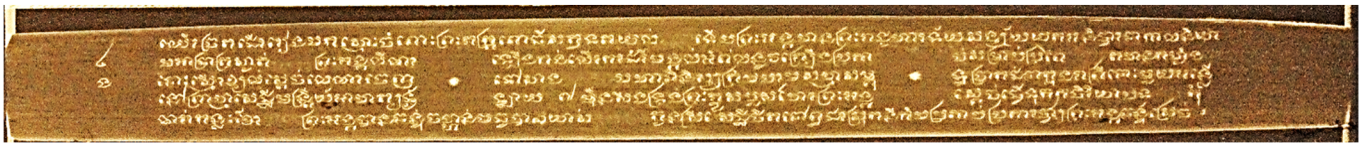

Figure 1. Sample of Khmer palm leaf manuscript with two holes used for binding.

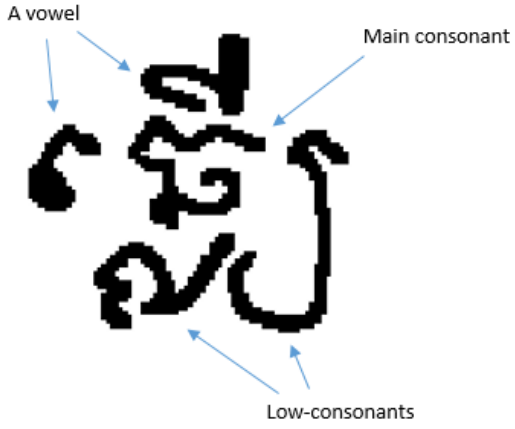

Figure 2. Example of combination of multi-level Khmer characters.

II. Related Work

Some of state-of-the-art methods to extract text lines from ancient and historical document images in [3] are based on the classic projection profile. The vertical profile is obtained by projecting and summing values of pixels or higher level units along each horizontal axis of the document. Text lines can be extracted by observing the peaks or valleys found in the profile. Nevertheless, the major drawback of this type of method is its sensitivity to the skew and fluctuation of text lines which often occur in handwritten documents. In [4], to make an improvement to the detection of slanted lines, the document is divided into vertical columns, and the projection profile from each column is calculated. The base of text lines in each column are extracted from the first row of gaps where there is no foreground pixel. The method is further improved in [5] and [6] by using smoothed profile of vertical columns. Baselines or local minima of the valleys of the profiles are instead used to define the starting location for separating text lines. These methods perform better in detecting skewed lines, but a post-processing stage is required to deal with incomplete or discontinued text lines.

Another category of line extraction techniques relies on geometrical information such as position, orientation, shape, and size of adjacent document units (blocks of foreground pixels or connected components) which are then grouped or merged together to form text lines [3,7]. Methods in this category are more adaptive to documents with complex layout; however, they tend to give unreliable results if there are irregularities, normally caused by noises and touching or overlapping of handwritten characters, in the topological structure of the document units.

Adiguzel, Sahin, and Duyguly [8] used a hybrid approach which combines aspects from both categories mentioned above. Other approaches to line extraction including smearing or blurring [9, 10], Hough transform [11], and seam carving [12, 13] have also been proposed.

Due to many challenges in text line segmentation, the problem still remains open especially for a specific type of documents such as Khmer palm leaf manuscripts. By taking advantage of the non-cursive characteristic of Khmer writing style, our approach manages to deal with multiple levels of Khmer characters and text lines with fluctuation and discontinuity in the palm leaf pages.

III. Line Segmentation Method

Our approach follows a bottom-up segmentation scheme. Fig. 3 presents an overview of our text line extraction system. The input data to the system are binary images of complete pages of palm leaf documents whose background is white and foreground is in black.

A. Connect Component Extraction

Connected components (CC) are extracted from the binary image by 8-connected neighborhood using a fast two-scan algorithm [14]. From all extracted components, we calculate the common width (W_{cc}) and common height (H_{cc}) using median width and height respectively. The coordinates of the center of mass of each CC are also noted.

B. Line Number Detection

The objective of this step of the system is primarily to detect the number of text lines in the manuscript page. Our method for line number detection is inspired by the Adaptive Partial Projection (APP) technique in [6]. The binary image of the document is first divided vertically into columns whose width is empirically set to $4W_{cc}$. Y-projection profile histogram is built from each column and then smoothed twice using moving average filter with window size of ⅓ H_{cc} and ½ H_{cc} respectively. The local maxima or peaks of the histogram of each column are considered to be the positions of median lines in that column. Spurious peaks are removed when their variance value is very small compared to the average variance of all peaks in the same column. The median value D_L of the distances between adjacent peaks is then calculated to filter peaks too close to each other. Suppose in column j, n peaks are detected. P_i represents the i^{th} peak from top to bottom ($0 < i < n$). P_i is removed if $P_i - P_{i-1} < ½\, D_L$. To avoid erroneous detection, we find the number of text lines N_L by choosing the 95$^{\text{th}}$ percentile of the list of number of peaks in each column sorted in ascending order.

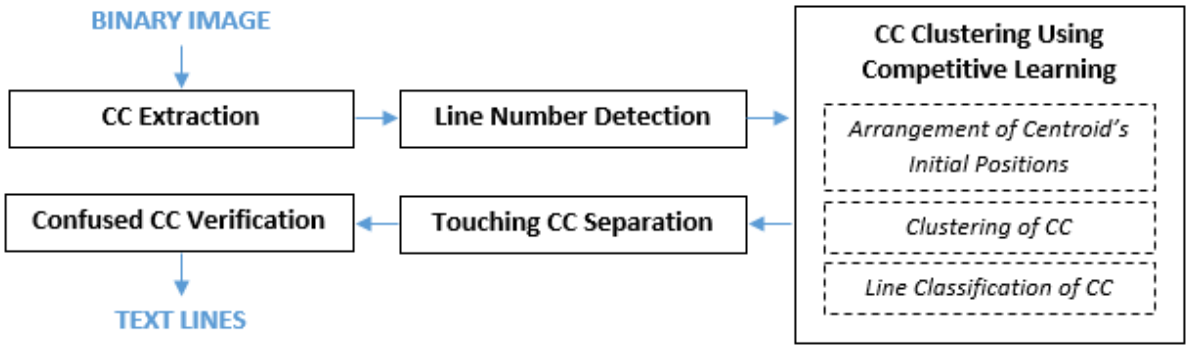

Figure 3. Overview of our line segmentation scheme.

C. *Clustering of CC's using Competitive Learning*

Competitive learning has been used widely for data clustering practically in many fields of study including image processing [15, 16]. The typical competitive learning algorithm takes an input data set and divides it into clusters. Each cluster is represented by its center point called centroid which changes adaptively from a pre-defined initial position. For each data point, the algorithm chooses among all centroids the one having the closest distance to be the winner. The winning centroid is then moved with a fraction of the distance towards the data point. The algorithm comes to a stop after a finite number of iterations normally when the moving fraction becomes zero or relatively small.

To segment the document image into text lines, we apply the competitive learning algorithm to a data set of one dimensional vectors consisting of the y-coordinates of the center of mass of all connected components. We note y_{C_i} the y-coordinate of the center of mass of a connected component $C_i, 0 \leq i < N_C$, with N_C the total number of connected components. We also note $Y_{m,n}$ $(0 \leq m < M$, with M the number of columns and $0 \leq n < N_L)$ the n^{th} centroid in column m. N_L centroids representing text line mid-points are placed at their initial positions in each column in the following manner:

- Find the starting column j whose y-projection histogram has exactly N_L peaks (with the highest average variance) and place one centroid at each peak.
- To the left of column j, one column at a time (column $k, k = j-1, j-2, \ldots$), find the peak closest to the corresponding peak of the column to its right (column $k+1$) and place a centroid there. If no peak is found, place a centroid at the same position of the corresponding centroid in column $k+1$. The distance between adjacent centroids in the same column is also maintained to be approximated to the common distance D_L (if the distance from the previously placed centroid $Y_{k,n-1}$ is less than ½ D_L, the new centroid $Y_{k,n}$ is instead placed at $Y_{k,n-1} + D_L$). The centroids are placed one by one from top to bottom in each column until the number N_L of centroids is reached. This placement can be expressed by the following pseudo-code:

for $k = j-1$ down to 0
  for $n = 0$ to $N_L - 1$
    $Y_{k+1,n'}$ closest peak to $Y_{k,n}$ at column $k+1$
    if $Y_{k+1,n'}$ is valid
      $Y_{k,n} = Y_{k+1,n'}$
    else
      $Y_{k,n} = Y_{k+1,n}$
    end if
    if $n > 0$ **and** $Y_{k,n} - Y_{k,n-1} < \frac{1}{2} D_L$
      $Y_{k,n} = Y_{k,n-1} + D_L$
    end if
  end for
end for

- To the right of column j, centroids are placed similarly one column at a time (column $j+1, j+2, \ldots$).

All centroids in each column are to be moved up or down by the adaptation rule stated below. For each component C_i, $Y_{m,n}$ wins if the following is satisfied:

$$\left|Y_{m,n} - y_{C_i}\right| \leq \left|Y_{m,r} - y_{C_i}\right| \; \forall r \in [0, N_L[\qquad (1)$$

The winning centroid is moved by:

$$Y_{m,n} \leftarrow Y_{m,n} + \alpha_k . \beta_{m,C_i} . (y_{C_i} - Y_{m,n}) \qquad (2)$$

For each iteration k, the fraction α_k converges to zero from an initial value ($\alpha_0 = 0.15$) so that the algorithm can halt:

$$\alpha_{k+1} = \frac{\alpha_k}{1+\alpha_k} \qquad (3)$$

In order to take into account also the x-distance between the component and the centroid, a weight β is used. If the

component is in the proximity of the centroid horizontally, the value of β is big, otherwise it becomes close to zero meaning that the component at a distance too far away will not have much effect on the movement of the centroid. For the value of β_m, we use a Gaussian function (mean μ is the x-coordinate of the centroid at column m) which is normalized so that $0 < \beta_m \leq 1$.

After the last iteration, the winning centroid of each connected component represents the text line that component belongs to. Fig. 4 shows how centroids are organized to adapt to the density of the connected components. The computational cost of the competitive learning algorithm depends linearly on the number of connected components in the document image.

D. Separation of Touching Connected Components

One of the issues in text line extraction is that some components touch each other to form a big blob which spreads over multiple lines. Such blobs have to be identified, separated, and classified into their correct line. To represent text line border points, a new set of points $Z_{m,n}$ ($0 \leq m < M$, M the number of columns and $0 \leq n < N_L - 1$) is defined as follows:

$$Z_{m,n} = \frac{Y_{m,n} + Y_{m,n+1}}{2} \quad (4)$$

A component C is considered to be spreading over line i if $y_{C,top} < Z_{m_C,i} - \gamma D_L$ or $y_{C,bot} > Z_{m_C,i-1} + \gamma D_L$ ($y_{C,top}$ and $y_{C,bot}$ are the y-coordinate of the top and bottom respectively of the bounding box of component C and by observation, coefficient γ is set to 0.2). If component C spreads over k lines (from line i to line $i + k - 1$), we divide it vertically into k new components using $Z_{m_C,j}$ ($i \leq j < i + k - 1$) as cutting locations. From top to bottom, the new divided components are assigned to line $i, i + 1, \ldots, i + k - 1$ respectively.

E. Confused Component Verification

As a consequence of multiple level characteristic of Khmer writing, certain characters are written far above or below its median text line. We note a normalized value $\Delta \in]-1,1[$ to represent how far a component is from its closest line. Fig. 5 illustrates this Δ value. A component C is called confused if $|\Delta_C| > T$, T some threshold (T is empirically set to 0.8 in our experiments). All confused components are reclassified to a new line either below or above it based on its relationship with adjacent components on that line. We look at its closeness with the components from neighboring lines. The confused component will be assigned to its other line (the line above it if $\Delta_C > 0$ or the line below it if $\Delta_C < 0$) if it has a shorter distance to the boundary of the nearest non-confused CC from that line compared to the distance to the nearest non-confused CC from its current line.

IV. Experimental Results

Our experiments are made on 50 document images (consisting of 224 lines) arbitrarily selected from EFEO database [2] for Khmer palm leaf manuscripts. Binary ground truths are created from those digitized images. When necessary, a local thresholding method [17] is applied, and noises caused by isolated pixels are then removed using median filter. The results are corrected with the help of photo editing software. The binarized document is superimposed on the original image, and strokes are traced manually using a stylus with pressure sensitive tip to maintain the variation of stroke width of each character in the manuscript. An example of the binary ground truths is shown in Fig. 6.

Some results of line segmentation using our approach on the binary ground truths are given in Fig. 7. The centroids or mid points form lines adaptive to skew, fluctuation, and discontinuity of the handwritten texts. Line border separating text lines can also be defined.

To evaluate and measure the performance of our algorithm, the ground truths for line segmentation are also developed using a semi-automatic tool (see Fig. 8) which generates a set of initial text line mid points. These points are then adjusted by being moved manually so that connected components are assigned to their correct line. Moreover, touching components are marked, and their vertical cutting locations are also noted.

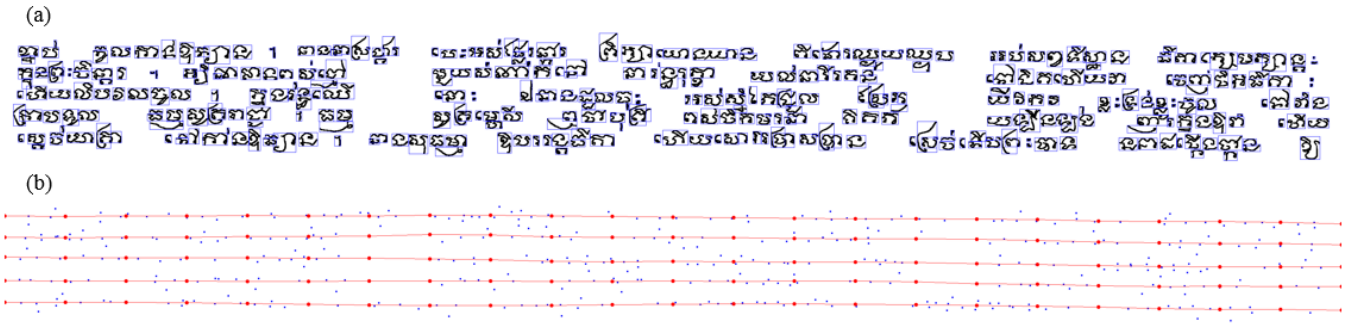

Figure 4. (a) Binarized image with connected components and their bounding boxes, (b) centroids in red are organized according to the density of connected components represented by their center of mass in blue.

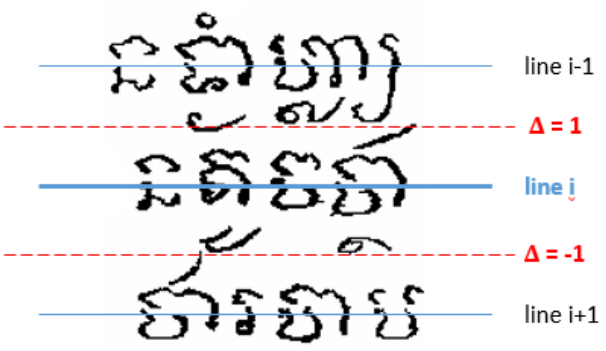

Figure 5. Δ value of components previously classified to line i.

To estimate our approach's accuracy, we compute the pixel level hit rate [18] and use F-measure as measurement. Suppose M lines are found in the ground truth, and N lines are detected using our proposed algorithm, P_i $(0 \leq i < M)$ and Q_j $(0 \leq j < N)$ represents the set of foreground pixels in the i[th] ground truth line and the j[th] detected line respectively. For each set Q_j, P_{k_j} is assigned to it if $P_{k_j} \cap Q_j$ contains the maximum number of foreground pixels:

$$|P_{k_j} \cap Q_j| \leq |P_i \cap Q_j| \ \forall i \in [0, M[\qquad (5)$$

The scores for precision, recall, and F-measure evaluated on each document page are computed as follows:

$$Precision = \frac{1}{N} \sum_{j=0}^{N-1} \frac{|P_{k_j} \cap Q_j|}{|Q_j|} \qquad (6)$$

$$Recall = \frac{1}{N} \sum_{j=0}^{N-1} \frac{|P_{k_j} \cap Q_j|}{|P_{k_j}|} \qquad (7)$$

$$F\text{-}measure = 2 \frac{Precision \times Recall}{Precision + Recall} \qquad (8)$$

The average matching-scores on documents in our data set are obtained. Table 1 illustrates these experimental results.

TABLE I. MATCHING-SCORES OBTAINED FROM OUR APPROACH

Precision (%)	Recall (%)	F-measure (%)
99.528	99.534	99.531

V. CONCLUSION

In this paper, we propose a text-line extraction method for Khmer ancient manuscripts written on palm leaves. Our approach uses piece-wise project profiles to detect line number and to set up initial positions of the centroids or text line mid-points. Those centroids are then moved adaptively to form correct lines by applying a clustering algorithm called competitive learning. Borders between text lines are defined so that they can be used as cutting locations to separate touching components that spread over multiple lines. Our method performs well on Khmer handwriting documents whose lines are often skewed or even fluctuated. It can also deal with discontinuity of text lines caused by holes made for document page binding. However, our method would need some adjustment so that it can be extended to extract text lines from other types of handwriting especially ones with cursive characteristic which produces connected components that are less in number and longer in width. One of the solutions is to divide connected components into shorter sections to guarantee enough data points before feeding them to our clustering algorithm. The approach to segment touching components can also be improved to take into account features of such components in order to find more efficient cutting paths.

ACKNOWLEDGMENT

This research study is supported by ARES-CCD (program AI 2014-2019) under the funding of Belgian university cooperation.

REFERENCES

[1] Z. Shi, S. Setlur, and V. Govindaraju. "Digital enhancement of palm leaf manuscript images using normalization techniques." In 5th International Conference On Knowledge Based Computer Systems, pp. 19-22. 2004.

[2] Khmer Manuscripts, École Française D'Extrême-Orient (EFEO). Retrieved from http://www.khmermanuscript.org

[3] L. Likforman-Sulem, A. Zahour, and B. Taconet. "Text line segmentation of historical documents: a survey." International Journal of Document Analysis and Recognition (IJDAR) 9, no. 2-4 (2007): 123-138.

[4] N. Tripathy, and U. Pal. "Handwriting segmentation of unconstrained Oriya text." Sadhana 31, no. 6 (2006): 755-769.

[5] M. Arivazhagan, H. Srinivasan, and S. Srihari. "A statistical approach to line segmentation in handwritten documents." In Electronic Imaging 2007, pp. 65000T-65000T. International Society for Optics and Photonics, 2007.

[6] R. Chamchong, and C. Fung. "Text line extraction using adaptive partial projection for palm leaf manuscripts from Thailand." In Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on, pp. 588-593. IEEE, 2012.

[7] A. Simon, J. C. Pret, and A. Johnson. "A fast algorithm for bottom-up document layout analysis." Pattern Analysis and Machine Intelligence, IEEE Transactions on 19, no. 3 (1997): 273-277.

[8] H. Adiguzel, E. Sahin, and P. Duygulu. "A Hybrid for Line Segmentation in Handwritten Documents." In Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on, pp. 503-508. IEEE, 2012.

[9] Y. Li, Y. Zheng, D. Doermann, and S. Jaeger. "Script-independent text line segmentation in freestyle handwritten documents." Pattern Analysis and Machine Intelligence, IEEE Transactions on 30, no. 8 (2008): 1313-1329.

[10] A. Nicolaou, and B. Gatos. "Handwritten text line segmentation by shredding text into its lines." In Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on, pp. 626-630. IEEE, 2009.

[11] G. Louloudis, B. Gatos, and C. Halatsis. "Text line detection in unconstrained handwritten documents using a block-based Hough transform approach." In Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on, vol. 2, pp. 599-603. IEEE, 2007.

[12] N. Arvanitopoulos, and S. Susstrunk. "Seam carving for text line extraction on color and grayscale historical manuscripts." In Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on, pp. 726-731. IEEE, 2014.

[13] X. Zhang, and C. Tan. "Text line segmentation for handwritten documents using constrained seam carving." In Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on, pp. 98-103. IEEE, 2014.

[14] L. He, Y. Chao, K. Suzuki, and K. Wu. "Fast connected-component labeling." Pattern Recognition 42, no. 9 (2009): 1977-1987.

[15] G. Budura, C. Botoca, and N. Miclău. "Competitive learning algorithms for data clustering." Facta universitatis-series: Electronics and Energetics 19, no. 2 (2006): 261-269.

[16] D. Rumelhart, D. Zipser, J. L. McClelland, et al. "Parallel Distributed Processing." Vol. 1. MIT Press. pp. 151–193, 1986.

[17] J. Sauvola, and M. Pietikäinen. "Adaptive document image binarization." Pattern recognition 33, no. 2 (2000): 225-236.

[18] O. Surinta, M. Holtkamp, F. Karabaa, J. Oosten, L. Schomaker, and M. Wiering. "A path planning for line segmentation of handwritten documents." In Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on, pp. 175-180. IEEE, 2014.

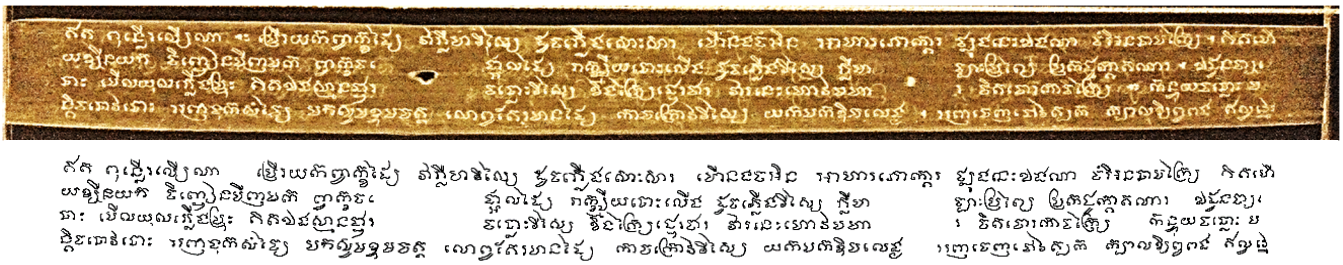

Figure 6. A palm leaf page (top) with its binary ground truth (bottom).

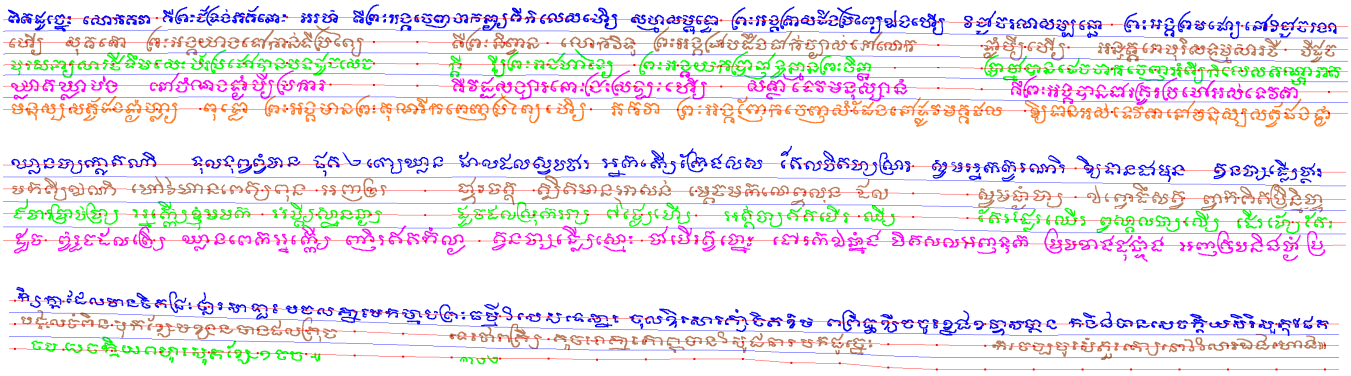

Figure 7. Sample results of our line segmentation method (median lines in red and borders of text lines in blue).

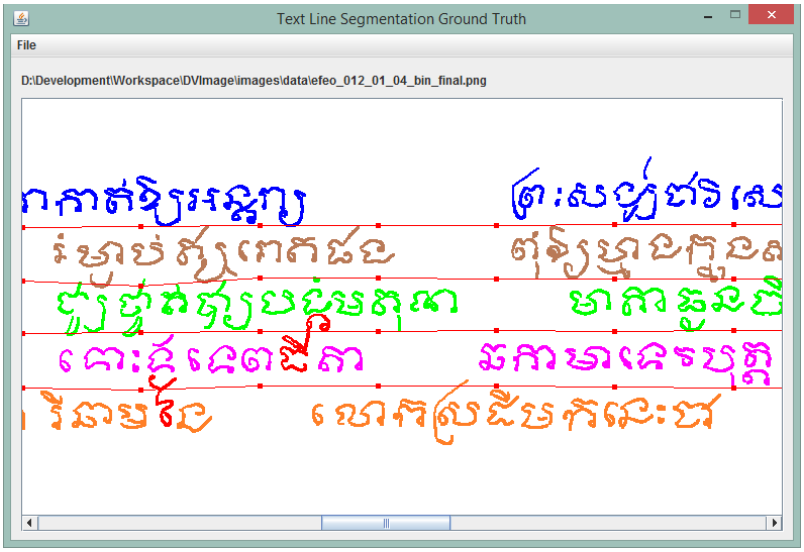

Figure 8. A tool to create ground truth for line segmentation (touching components are marked red.