# Line Segmentation for Grayscale Text Images of Khmer Palm Leaf Manuscripts

Dona Valy[1,2], Michel Verleysen[1], and Kimheng Sok[2]
[1]ICTEAM Institute, Université Catholique de Louvain, Belgium
[2]Department of Information and Communication Engineering, Institute of Technology of Cambodia, Cambodia
{dona.valy,michel.verleysen}@uclouvain.be, sok.kimheng@itc.edu.kh

***Abstract*— Text line segmentation is one of the most essential pre-processing steps in character recognition and document analysis. In ancient documents, a variety of deformations caused by aging produce noises which make the binarization process very challenging. Moreover, due to the irregular layout such as skewness and fluctuation of text lines, segmenting an ancient manuscript page into lines still remains an open problem to solve. In this paper, we propose a novel line segmentation scheme for grayscale images of Khmer ancient documents. First, a stroke width transform is applied to extract connected components from the document page. The number and medial positions of text lines are estimated using a modified piece-wise projection profile technique. Those positions are then modified adaptively according to the curvature of the actual text lines. Finally, a path finding approach is used to separate touching components and also to mark the boundary of the text lines. Experiments are conducted on a dataset of 110 pages of Khmer palm leaf manuscript images by comparing the robustness of the proposed approach with existing methods from the literature.**

***Keywords*— handwritten document analysis; text line segmentation; palm leaf manuscript.***

## I. INTRODUCTION

Historical handwritten documents are of important cultural value and are a source of useful information and knowledge from earlier times. Likewise palm leaf manuscripts are considered to be one of the national heritages of Cambodia. For the purpose of preservation and in order to turn old palm leaf documents into digital format, automatic text recognition system has to be involved so that the contents of those documents become easily accessible. One of the key steps in the pipeline of text recognition system is the extraction of text lines since incorrectly segmented text lines can greatly influence the final output of the recognition.

One of the main difficulties in extracting text lines from Khmer palm leaf documents is caused by the characteristic of Khmer writing and how words are formed. 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. These variations of how Khmer letters are formed produce multiple levels (sometimes more than three) of characters. Some writers also tend to exaggerate their writing by elongating the upper or lower part of a character which makes it go far out of its main line, touch, or overlap with other characters from adjacent lines. Moreover, due to high width-height ratio of the manuscript page, text lines are written very close to each other with very little spacing between them. Because text lines are long, they may also be slanted, curved upward/downward, or fluctuated on account of them being handwritten. In addition, strings used to tie palm leaves together to create a book-like binding leave behind holes surrounded by empty areas in the middle of the page producing discontinuity of text lines. Fig. 1 shows some sample images of digitized palm leaf document pages illustrating those challenges.

Some distinct properties of Khmer palm leaf documents can also be noted. The manuscript pages are made from dried palm leaves with light yellowish or brown color. Ancient Khmer characters are carved on each page, and a mixture of coal powder and a type of paste is applied on the carving resulting in dark black text. We can therefore be sure that the documents contain dark color foreground text over light color background. Furthermore, despite its form being elaborative and complex, Khmer handwriting is not cursive. Connected components representing individual characters can be extracted from each text line.

In this paper, we propose a segmentation approach focusing on line extraction from grayscale images of ancient Khmer palm leaf manuscripts. The method manages to deal with challenges and difficulties of Khmer handwriting scripts taking into account its distinct properties.

The organization of the rest of the paper is as follows: Section II briefly discusses existing literature on line segmentation on both binary and grayscale images. Section III describes the detailed overview of the proposed line extraction

Figure 1. Sample digitized images of Khmer palm leaf manuscripts

978-1-5386-1842-4/17/$31.00 ©2017 IEEE

technique. Section IV presents the experimental results and performance evaluation of the proposed method compared to existing approaches in the literature. Finally, a conclusion is given in Section V.

## II. RELATED WORK

Numerous state of the art methods of line segmentation have been proposed. However, many of them operate only on binary images. In [1-4], a variety of projection profile techniques are used. The goal of this type of methods is to extract estimated medial or seam positions of text lines based on the peaks/valleys of the histogram profiles projected on vertical axis of the document page. Another genre of line segmentation approaches tries to find optimal paths which pass through text line seams [5-8]. Tracers following the white-most and black-most trails are used in [5] in order to shred the image into text line areas. In [6], an improved Viterbi algorithm based on Hidden Markov Model generates all possible paths, and a filtering process is applied afterwards to remove invalid paths leaving behind only the optimal ones. To select the ideal paths, the approach in [7] computes an energy map which accumulates energy from left to right for each path while a traveling cost function of the path is instead calculated in [8]. Other methods of text line segmentation working on binary images include smearing/blurring [9], Hough transform [10], and water flow [11]. For the approaches mentioned above, a good initial binarization process is required. However, common degradations and noises on aging palm leaves such as seepage of ink or bleed through, tears, stains from dirt and moisture, and other types of damage render this task impossible to produce an acceptable result. It is therefore necessary that a new binarization-free scheme is developed.

Some methods which are independent on the binarization process have also been proposed. In [12], projection profile is applied directly on a grayscale document. The profile is computed by summing the pixel values from each row. Different skew angles can be detected and estimated. However, one of the drawbacks from the method is that seams separating text lines are straight along the direction of the detected skew. This is not suitable for documents with little spacing between adjacent lines whose text components may touch or be too close to each other that the seams are not able to separate them correctly. Another technique transform the document page into a set of interest points which can be grouped into clusters to form text lines based on their local orientation [13]. The approach of [14] makes use of seam carving technique to compute separating seams between text lines by constraining the optimization procedure inside a defined region. Similarly, in [15], areas for segmentation path are selected, and the multistage graph search algorithm is applied to find the shortest nonlinear path in each area.

Due to specific characteristics of Khmer handwriting, text line extraction from Khmer palm leaf manuscripts still remains an open problem. By taking into account some properties of this type of document such as its non-cursive writing style, a performance improvement over state of the art methods is expected. The new proposed scheme should manage to deal with challenges including detecting skew/fluctuation of text lines and especially the discontinuity of text lines which is difficult to solve for many of the existing approaches.

## III. TEXT LINE SEGEMENTATION

An overview pipe line of the proposed text line segmentation approach is presented in Fig. 2. The input data to the system is a grayscale image of a palm leaf manuscript page.

### *A. Extracting Connected Components*

In order to extract text components from a document page, edges in the image are first calculated using Canny edge detection method [16]. A text localization approach called Stroke Width Transform (SWT) [17] is then applied. The approach creates a new feature map storing the width values of the most likely strokes containing each pixel. We then group together adjacent pixels with similar stroke width into characters (connected components). The average stroke width $SW_{avg}$ of the whole stroke map is also computed.

Unwanted noise components such as big patches, long horizontal or vertical strokes, and small dots are removed using the following rules:

- The mean of stroke width values of all pixels in the connected component must be less than $2.SW_{avg}$.
- The length-width and width-length ratio of the component must be less than 3.
- The length or width of the component must be greater than $3.SW_{avg}$.

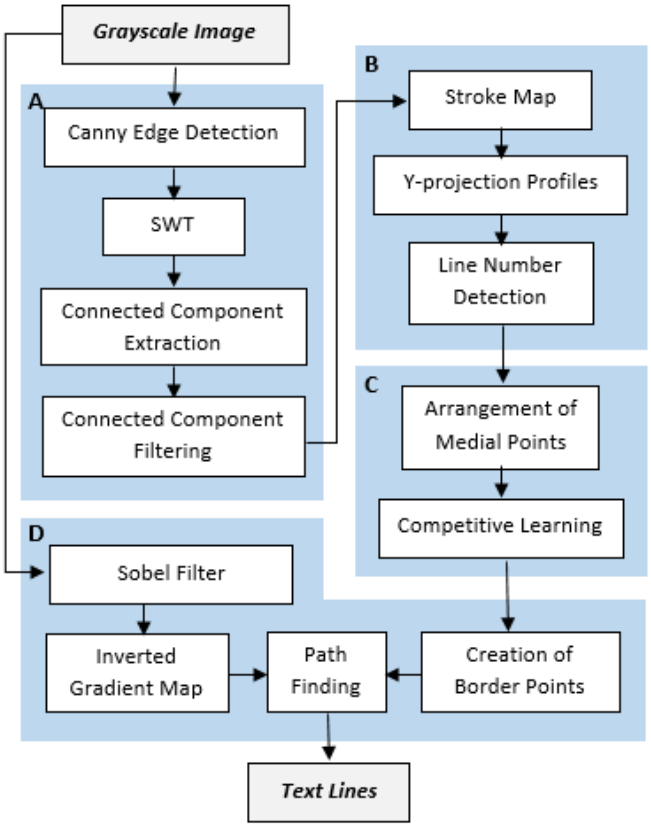


Figure 2. Overview of the proposed line segmentation pipe line

After all noise components are filtered out, we obtain a new stroke map $I_{CC}$ consisting of all stroke pixels as black foreground over a white background (Fig. 3). The median height $H_{median}$ of all connected components is also calculated.

### B. Detecting Number of Text Lines

The main goal of this step is to determine the number of text lines in the manuscript page. We extend the method in our previous work [18] by applying it on the $I_{CC}$ map obtained from the previous step. First, X-projection profile is built from the map to find the starting and ending positions of the text region. The discovered region is divided into $N_C$ vertical columns whose width is empirically set to $W_C = 10.H_{median}$. We then construct a Y-projection profile histogram for each column, and we smooth them twice using moving average filters with widow size of $1/3H_{median}$ and $1/2H_{median}$ respectively. We consider the local maxima or peaks of the smoothed histogram of each column to be the medial line positions in that column. The variance value of each peak is computed, and spurious peaks with very small value of variance compared to the average variance values of all peaks in their same column are removed. To further filter out incorrect peaks, we also remove peaks too close to each other. Suppose in a column, $n$ peaks are detected. $P_i$ represents the $i^{th}$ peak from top to bottom $(0 < i < n)$. We remove $P_i$ if the distance between $P_i$ and $P_{i-1}$ is less than $1/2D_{median}$ where $D_{median}$ is the median value of all distances between two adjacent peaks. The final numbers of peaks in all columns are stored in a sorted list from which we choose the value in the 95th percentile to be the number of text lines $N_L$ in the document page.

### C. Adapting Text Lines to Skew/Fluctuation

To construct an estimation of medial seam of a text line, we connect corresponding medial points situated at the peaks of the Y-projection histogram of neighboring columns together. However, some peaks may be missing due to the fact that different numbers of peaks can be found in different columns. To guarantee a smooth connection between peaks, new medial points may be added. We want to have eventually a $N_L \times N_C$ grid of medial points. Let's denote $M_{l,c}$ a medial point on text line $l$ in column $c$ with $0 \le l < N_L$ and $0 \le c < N_C$ and $M_{l,c}^x, M_{l,c}^y$ its x-coordinate and y-coordinate respectively. Each medial point can be added as follows. First, relative to the starting point of the text region, each medial point should be placed horizontally at the center of its column ($M_{l,c}^x = c.W_C + W_C/2$) . Then we select a starting column $j$ whose Y-projection profile consists of exactly $N_L$ peaks. Among multiple instances of such columns, we pick the one with the highest average peak variance and place one medial point at each peak. To the left of column $j$, we iterate one column at a time (column $k$,$k = j-1, j-2, \dots, 0$) to find the closest peaks to the corresponding peaks of the column to its right (column $k+1$) and place a medial point there. If no peak is found, we place a medial point at the same y-coordinate position of the corresponding medial point in column $+1$ ($M_{l,k}^y = M_{l,k+1}^y$). We also maintain the distance between two adjacent medial points from the same column to be approximately equal to the common distance $D_{median}$. To do so, if the distance from the previously placed medial point $M_{l-1,k}$ is less than $1/2D_{median}$, the new medial point $M_{l,k}$ is instead placed at $M_{l,k}^y = M_{l-1,k}^y + D_{median}$. We place the medial points one by one from top to bottom until we reach the total number of $N_L$ points per column. The placement can be expressed by the pseudo code shown in Algorithm 1. Similarly, to the right of column j, medial points are added from top to bottom one column at a time (column $k$, $k = k+1, k+2, \dots, N_C - 1$).

**for** $k = j-1$ down to 0
  **for** $l = 0$ to $N_L - 1$
    place $M_{l,k}$ at the peak closest to $M_{l,k+1}$
    **if** $M_{l,k}$ is NOT valid
      $M_{l,k}^y = M_{l,k+1}^y$
    **end if**
    **if** $l > 0$ **and** $M_{l,k}^y - M_{l-1,k}^y < \frac{1}{2} D_{median}$
      $M_{l,k}^y = M_{l-1,k}^y + D_{median}$
    **end if**
  **end for**
**end for**

Algorithm 1. Pseudo code illustrating the placement of medial points

While the above procedure already gives an acceptable medial seam of text lines it is important to further adapt the seams accurately to the skew and fluctuation of text lines. For that purpose a competitive algorithm is used in order to adapt the vertical position of medial points to the surrounding connected components. Let us denote $y_{C_i}$ the y-coordinate of the center of mass of a connected component $C_i, 0 \le i < N_{CC}$ where $N_{CC}$ is the total number of connected components extracted. The grid-like set of medial points extracted as above plays the role of centroids for the competitive learning algorithm. The so-called winning centroid with respect to each component $C_i$, is the medial point $M_{l,c}$ whose vertical distance to $C_i$ is the smallest:

$$|y_{C_i} - M_{l,c}^y| \le |y_{C_i} - M_{r,c}^y| \ \ \forall r \in [0, N\_L[ \qquad (1)$$

After being selected as a winner, the winning medial point is moved by a fraction of its distance towards the connected component $C_i$.

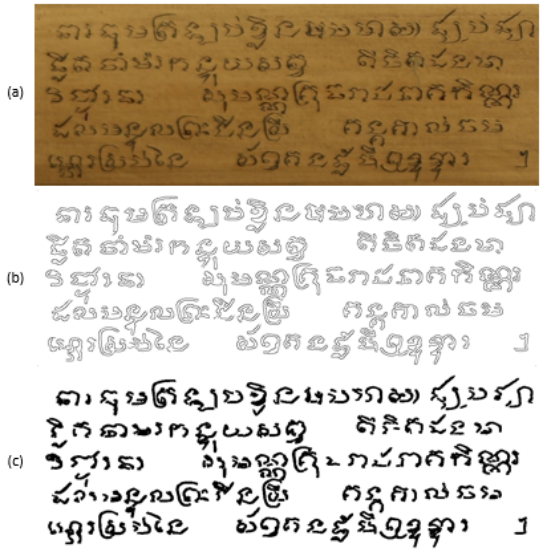


Figure 3. (a) Original Image, (b) Edge map using Canny edge detection, (c) Stroke map

$$M_{l,c}^{y} \leftarrow M_{l,c}^{y} + \alpha_k . \beta_{c,C_i} . \left(y_{C_i} - M_{l,c}^{y}\right) \quad (2)$$

After each iteration k, the learning step $\alpha_k$ is forced to decrease towards zero ($\alpha_k = \alpha_{k-1}/(1+\alpha_{k-1})$) from an initial value $\alpha_0$ ($\alpha_0$ is set to 0.15 in our experiment) so that we can halt the algorithm. In order to take into account the horizontal distance between the component $C_i$ and medial point, we introduce a weight $\beta$: the value of $\beta$ is large if the connected component is close to the medial point, and decreases with the distance to the latter. This means that the components at a distance far away do not have much effect on the movement of the medial point. We use a Gaussian function (mean $\mu = M_{l,c}^{x}$) for the value of $\beta_c$ which is normalized so that $0 < \beta_c \leq 1$.

### D. Creating Boundaries between Text Lines

To define the boundary between two adjacent lines, a path finding technique is used. Our path finding approach is inspired by the A* path planning method proposed by Surinta et al [8]. The objective of path planning is to compute the shortest path from a starting point to its destination avoiding obstacles along the way. A* is one of the path planning algorithms that minimizes the traveling costs between states (a state refers to each pixel in the sequence of neighboring pixels representing a particular path) from the starting state to the goal state. To solve the line segmentation problem, paths separating text lines need to be traced from the left side (starting state) to right side (goal state) of the text, and the pixels which belong to text are viewed as obstacles. However due to some handwritten text components from adjacent lines being touched or over- lapped, the goal state can be unreachable. A modified A* path-planning technique is proposed here to allow the path to pass through such components.

First, separating seams between text lines are created from the already defined medial points. We denote a new set of border points $B_{l,c}$, $0 \leq l < N_L - 1$ and $0 \leq c < N_C$. It is defined as follows:

$$B_{l,c} = \frac{M_{l,c} + M_{l+1,c}}{2} \quad (3)$$

For each separating seam, we find a sequence of $N_c - 1$ paths, and each path starts from the starting state $s_1$ at $B_{l,k}$ to the goal state $s_n$ at $B_{l,k+1}$, $k \in [0, N_C - 1[$. To emphasize the text foreground, the original grayscale image of the manuscript page is filtered using Sobel operator. The inverted gradient map $I_{grad}$ of the image is used so that the pixel intensity of the foreground text is less than the one of its background. Two cost functions are computed and combined to obtain the final traveling cost $C(s_i, s_j)$ between states. If we denote $s_{1,a}, s_{2,a}, \ldots, s_{n_a,a}$ as the sequence of states traversed by path $p_a$, the goal of the path finding algorithm is to determine the optimal path p_optimal with the minimum total traveling cost:

$$p_{optimal} = \arg\min_{p_a} \sum_{i=1}^{n_a - 1} C(s_{i,a}, s_{i+1,a}) \quad (4)$$

The two function costs are described as follows:

#### 1) Intensity difference cost function $D(s_i, s_j)$:

This function enforces the path to avoid passing through foreground pixels. The function returns a large value when the pixel intensity abruptly changes from high to low demonstrating a situation when an edge of a foreground text stroke is encountered (going from background pixels to foreground pixels). The function however does not give penalty but encourages it when the path leaves the foreground stroke (going from foreground pixels to background pixels). The cost function $D(s_i, s_j)$ is computed by:

$$D(s_i, s_j) = \max(D_I . |D_I|, 0) \quad (5)$$

$$\text{where } D_I = I_{grad}(s_i^x, s_i^y) - I_{grad}(s_j^x, s_j^y) \quad (6)$$

#### 2) Vertical distance cost function $V(s_j)$:

This function reassures that the path does not deviate too far from the seam line preventing the path from jumping up or down the entire line. It is the vertical distance from the state $s_j$ to the slanted seam line constructed from the two points at $s_1$ and $s_n$. This cost function is defined as:

$$V(s_j) = \left| s_j^y - s_1^y - \frac{(s_j^x - s_1^x)(s_n^y - s_1^y)}{s_n^x - s_1^x} \right| \quad (7)$$

In order to connect continuously consecutive paths, we set the starting state $s_1$ of the next path in the sequence to be at the same position as the goal state $s_n$ from the previous path.

We combine these two cost functions to achieve the final traveling cost $C(s_i, s_j)$ defined as follows:

$$C(s_i, s_j) = c_d . D(s_i, s_j) + V(s_j) \quad (8)$$

where $c_d$ is a parameter which controls how the path choose its next state to traverse. A large value of c_d means that the path would prefer staying far away from the seam line rather than passing through foreground text pixels.

Since the path traverses through states from left to right, instead of computing the cost functions for neighbor pixels in all eight directions, only five directional steps are considered: South, South-East, East, North-East, and North.

## IV. EXPERIMENTAL RESULTS

### A. Dataset and Text Line Segmentation Ground Truth

We conduct experiments on a new dataset of Khmer palm leaf manuscripts called SleukRith Set. The set consists of 110 pages of digitized manuscript images randomly selected from existing digital collections obtained from EFEO database[1], the National Library[2], and the Buddhist Institute [3]in addition to the recently digitized images by our own digitization campaign in various regions in Cambodia. The dataset contains ground truth information which includes three types of data: isolated characters, words, and lines.

In order to segment and annotate a manuscript page into small image patches representing each individual character, a

[1] http://khmermanuscripts.efeo.fr
[2] https://www.facebook.com/NLC.gov.kh
[3] https://www.budinst.gov.kh

polygon boundary enclosing the character needs to be drawn manually. The human ground truther is required to dot out vertices of the polygon one by one until a proper boundary is formed. After all characters in the page are manually annotated, they may be grouped together and be inserted into a line. The union of the polygon areas of all isolated characters in a line represents the total region of that line. Fig. 4 shows the tool used to create line annotation.

### B. *Evaluations and Results*

In order to evaluate the performance of the proposed method and also to compare it with existing methods in the literature, we adopt the measures used in the ICDAR2013 Handwriting Segmentation Competition [20]. According to the evaluation method, we count the number of one-to-one matches between text lines detected by the approach and the text lines in the ground truth. The matching score is computed as follows:

$$MatchScore(i,j) = \frac{T(G_j \cap R_i \cap I_{IC})}{T\left((G_j \cup R_i) \cap I_{IC}\right)} \quad (9)$$

where $T(s)$ is function that counts the number of points in set $s$; $G_j$ is the set of all points inside the union of all polygon regions of isolated characters in the ground truth belonging to text line $j$; $R_i$ the set of all points inside the region of result text line $i$; and $I_{IC}$ the set of all points inside the union of all polygon regions of ground truth isolated characters in the whole document page.

We consider a region pair to be a one-to-one match only if the matching score is above an acceptance threshold $T_a$. Let's assume $N$ to be the number of text lines found in the ground truth, $M$ to be the number of detected text lines by the approach, and $o2o$ to be the number of one-to-one match pairs, then the detection rate ($DR$) and recognition accuracy ($RA$) are defined as:

$$DR = \frac{o2o}{N} \quad (10)$$

$$RA = \frac{o2o}{M} \quad (11)$$

By combining $DR$ and $RA$, we obtain the evaluation metric F-measure score $FM$:

$$FM = \frac{2.DR.RA}{DR+RA} \quad (12)$$

Among the 110 pages of SleukRith Set, 10 pages are randomly selected and are used for parameter tuning (the optimal value of $c_d = 3$ is found). We then apply the proposed approach to the other 100 pages (the total number of text lines in the ground truth $N = 476$). For comparison, the same subsets are also used for the methods in [14] and [15]. The evaluation results using the acceptance threshold $T_a = 0.9$ are illustrated in Table 1. As it can be observed from Table. 1, the proposed method outperforms the other approaches by a large margin. Some results of line segmentation using the proposed method dealing with skew, fluctuation, and discontinuity of text lines are given in Fig. 6.

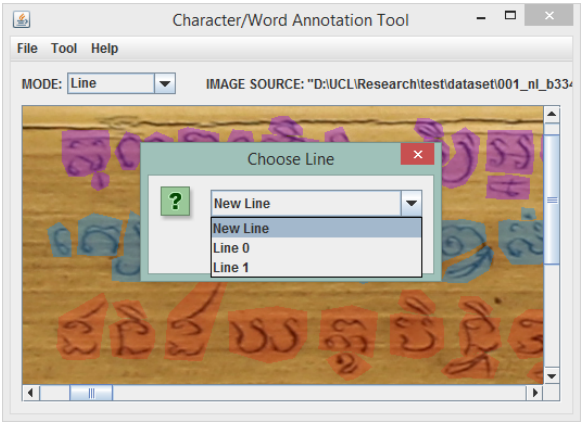


Figure 4. A tool used to construct line segmentation ground truth

TABLE I. RESULT OF PERFORMANCE EVALUATION

| Method | M | o2o | DR (%) | RA (%) | FM (%) |
|---|---|---|---|---|---|
| Arvanitopoulos et al. [14] | 665 | 356 | 53.53 | 74.79 | 62.40 |
| Kesiman et al. [15] | 505 | 157 | 31.09 | 32.98 | 32.01 |
| Proposed | 484 | 446 | 92.15 | 93.70 | 92.92 |

## V. TEXT LINE SEGEMENTATION

In this paper, a line segmentation scheme is proposed for Khmer palm leaf manuscripts. The approach is binarization free and works directly on grayscale images. First, we apply a stroke width transform (SWT) on the edge map of the image to extract connected components. We then determine the number and medial positions of text lines which are to be modified adaptively according to skew and fluctuation of the actual text lines. Finally, we construct text line boundaries using a path finding technique. We conduct experiments on a dataset of 110 pages of Khmer palm leaf manuscript images, and the evaluation results demonstrate that the approach presented in this paper produces a very promising outcome compared to other existing methods in the literature.

In future work, this line segmentation scheme can be included in a handwritten text recognition pipeline in conjunction with popular machine learning techniques such as LSTM and other recurrent neural networks.

## ACKNOWLEDGMENT

The authors would like to thank the National Library of Cambodia, the EFEO team, and the Buddhist Institute for providing their digital images of palm leaf manuscripts. In addition, we would also like to acknowledge the help with the annotation process of our dataset by volunteer students from the Institute of Technology of Cambodia (ITC) and the National Institute of Posts, Telecommunications, and ICT (NIPTICT). This research study is supported by ARES-CCD (program AI 2014-2019) under the funding of Belgian university cooperation.

## REFERENCES

[1] Tripathy, N., & Pal, U. (2006). Handwriting segmentation of unconstrained Oriya text. Sadhana, 31(6), 755-769.

[2] Ptak, R., Żygadło, B., & Unold, O. (2017). Projection-based text line segmentation with a variable threshold. International Journal of Applied Mathematics and Computer Science, 27(1), 195-206.

[3] Arivazhagan, M., Srinivasan, H., & Srihari, S. N. (2007, January). A statistical approach to line segmentation in handwritten documents. In DRR (p. 65000T).

[4] Chamchong, R., & Fung, C. C. (2012, September). 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.

[5] Nicolaou, A., & Gatos, B. (2009, July). 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.

[6] Peng, G., Yu, P., Li, H., & He, L. (2016, July). Text line segmentation using Viterbi algorithm for the palm leaf manuscripts of Dai. In Audio, Language and Image Processing (ICALIP), 2016 International Conference on (pp. 336-340). IEEE.

[7] Zhang, X., & Tan, C. L. (2014, September). 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.

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

[9] Li, Y., Zheng, Y., Doermann, D., & Jaeger, S. (2008). Script-independent text line segmentation in freestyle handwritten documents. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(8), 1313-1329.

[10] Louloudis, G., Gatos, B., & Halatsis, C. (2007, September). 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.

[11] Brodić, D., & Milivojević, Z. N. (2016). Text line segmentation with the parametric water flow algorithm. Information Technology And Control, 45(1), 52-61.

[12] Bar-Yosef, I., Hagbi, N., Kedem, K., & Dinstein, I. (2009, July). Line segmentation for degraded handwritten historical documents. In Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on (pp. 1161-1165). IEEE.

[13] Garz, A., Fischer, A., Bunke, H., & Ingold, R. (2013, August). A binarization-free clustering approach to segment curved text lines in historical manuscripts. In Document Analysis and Recognition (ICDAR), 2013 12th International Conference on (pp. 1290-1294). IEEE.

[14] Arvanitopoulos, N., & Süsstrunk, S. (2014, September). 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.

[15] Kesiman, M. W. A., Burie, J. C., & Ogier, J. M. (2016, October). A New Scheme for Text Line and Character Segmentation from Gray Scale Images of Palm Leaf Manuscript. In Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on (pp. 325-330). IEEE.

[16] Canny, J. (1986). A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence, (6), 679-698.

[17] Epshtein, B., Ofek, E., & Wexler, Y. (2010, June). Detecting text in natural scenes with stroke width transform. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on (pp. 2963-2970). IEEE.

[18] Valy, D., Verleysen, M., & Sok, K. (2016, October). Line Segmentation Approach for Ancient Palm Leaf Manuscripts Using Competitive Learning Algorithm. In Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on (pp. 108-113). IEEE.

[19] Budura, G., Botoca, C., & Miclău, N. (2006). Competitive learning algorithms for data clustering. Facta universitatis-series: Electronics and Energetics, 19(2), 261-269.

[20] Stamatopoulos, N., Gatos, B., Louloudis, G., Pal, U., & Alaei, A. (2013, August). ICDAR 2013 handwriting segmentation contest. In Document Analysis and Recognition (ICDAR), 2013 12th International Conference on (pp. 1402-1406). IEEE.

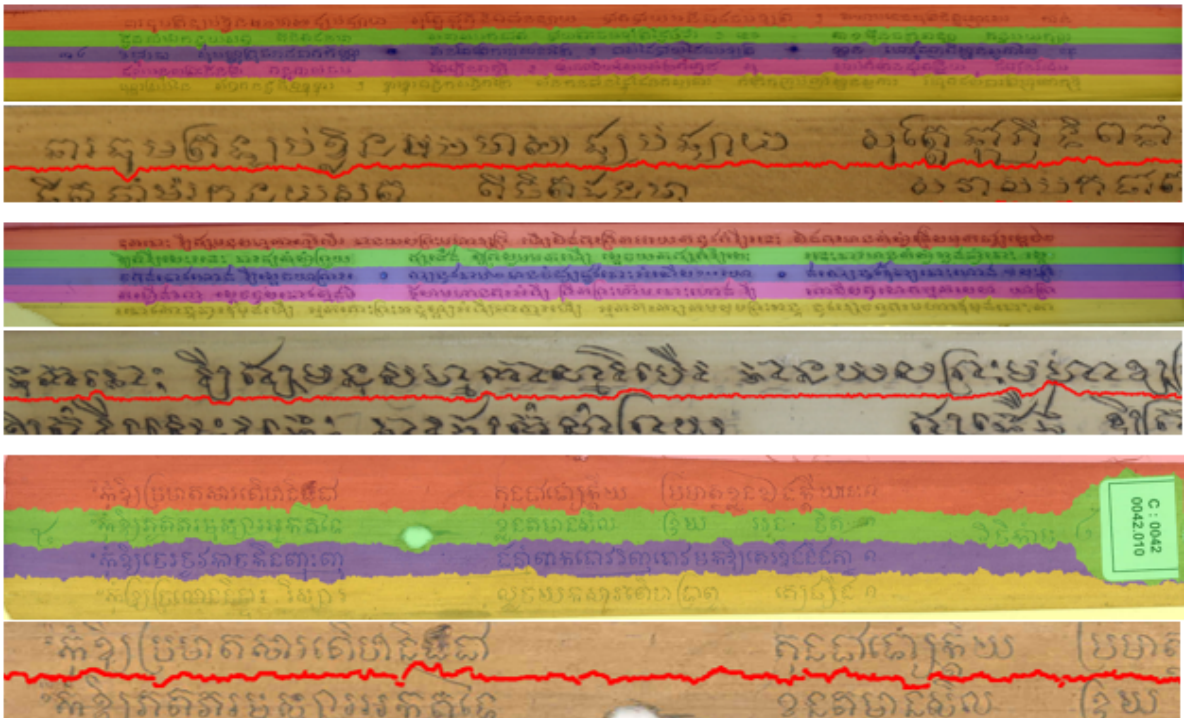

Figure 5. Segmentation results of the proposed approach (pairs of whole segmented manuscript page and zoomed out area with medial seams marked in red)