Improving Projection Profile for Segmenting Characters from Javanese Manuscripts

Aditya W Mahastama[1], Lucia D Krisnawati[1]
[1]*Informatics Dept., Faculty of Information Technology, Universitas Kristen Duta Wacana, Jl. Dr. Wahidin Sudirohusodo 5-25, Yogyakarta, Indonesia*

Keywords: OCR, Character Segmentation, Projection Profile Cutting, Outlier Analysis.

Abstract: The emergence of non-latin scripts in the Unicode character set has opened the possibilities to do Optical Character Recognition (OCR) for manuscripts written in non-alphabetic scripts. Javanese is one of the Southeast Asian languages which has vast collections of manuscripts. Unfortunately, these manuscripts are prone to damage due to lack of maintenance. Therefore, digitising them through OCR has become the most obvious option. This research focuses on the segmentation process of our OCR project which implements the Projection-Profile Cutting (PPC). The rationale is that PPC is well known as having a low computational cost. As the object of segmentation, we sampled 72 scanned pages of Serat Mangkunegara IV, Wulang Maca, and Kitab Rum. Our preliminary evaluation showed that implementing PPC per se exhibits unsatisfactory results. Hence, we refined it by applying a statistical analysis to segment lines of characters whose distance is too low. The proposed algorithm results in 19.112 segments. To evaluate the system outputs, we conducted two levels of evaluation: the line and character segmentations. The refinement of PPC has proved to increase the line segmentation accuracy by 32.84%. To evaluate the character segmentation, we collaborated with Javanese Wikipedia Community which verified them manually in 4 batches. Only 15.386 segments were verified, in which 73.59% (11.322) system outputs are correctly segmented, 22.5% (3.464) are over-segmented, 1.3% (206) are under-segmented, and the rest has not been labelled as either one of three categories above.

1 INTRODUCTION

Character segmentation is mutually inseparable from text segmentation. Both text and character segmentations frequently become a pre-processing stage in a more complex applications such as in a paper watermarking system (Mei, et al., 2013), processing bank checks and recognizing mails (Lacerda & Mello, 2013), writer identification in historical documents (Dhali, et al., 2017), or in a vehicle license plate detection system (Asif, et al., 2017). Character segmentation has turned to be a significant phase in Optical Character Recognition (OCR) since it could increase as well as decrease its recognition rate.

The Optical Character Recognition is not merely applied for the academic purposes such as digitizing primary sources for the sake of building a digital library (Manmatha & Rath, 2003). In industrial area, OCR is applicable to detect and recognize the running texts in a video (Chaiwatanaphana, et al., 2017); its software is frequently embedded as additional features to scanners, printers or smartphones in order to increase their sale. This research focuses on the character segmentation as a subtask of OCR for digitizing and preserving the content of the historical manuscripts.

In OCR, character segmentation refers to a process of separating the pixels of a text image from the pixels of its background. The text image is acquired by either scanning or photographing books or manuscripts. The process proceeds to segment lines of characters, aka horizontal segmentation, and to decompose each segmented line vertically into individual character images (vertical segmentation). These characters will be the input of the recognition process.

In this study, we investigate how to segment Javanese characters acquired from scanning the historical books. Javanese script is categorized as an Abugida, which is a writing system where the consonant letters represent syllables with a default

Mahastama, A. and Krisnawati, L.
Improving Projection Profile for Segmenting Characters from Javanese Manuscripts.
DOI: 10.5220/0008526900770082
In *Proceedings of the 1st International Conference on Intermedia Arts and Creative Technology (CREATIVEARTS 2019)*, pages 77-82
ISBN: 978-989-758-430-5

vowel and other vowels are denoted by diacritics (Ding, et al., 2018). In Javanese, diacritics could be written above, under, on the left or right side of the main characters. Hence, its segmentation challenges lie not only on the overlapped character strokes and diacritics but also on the identification which diacritic belongs to which character. In our preliminary study, we have experimented segmentation by Projection Profile (PP). However, the application of Projection Profile per se turned out to give unsatisfactory results. Therefore, we proposed to refine it by means of detecting outlier of pixel density. The hybrid of pixel outlier detection and PP has proved to increase the line segmentation accuracy by 32.84%.

2 RELATED WORKS

Various character segmentation methods have been proposed in OCR and in many text recognition among images (Asif, et al., 2017; Karthikeyan, et al., 2013). However, Projection Profile methods and Connected Component Analysis (CCA) are among those which are frequently applied due to their low computational cost and simplicity. The application of these methods have technically slight differences owing to the text media, for example texts written on palm leaves (Kesiman, et al., 2016), books/manuscripts (Mei, et al., 2013), or car license plates (Karthikeyan, et al., 2013), and the characteristics of the characters themselves. Apart from applying PP and PCA which base their segmentation on the pixel intensity, some researches make use of character features as a basis of segmentation as found in (Budhi & Adipranata, 2015; Lacerda & Mello, 2013), or a hybrid between feature-based and Projection Profile as in (Lue, et al., 2010). A more intricate approach is the recognition-based segmentation which adapts prior knowledge to screen all possible segmentation schemes (Mei, et al., 2013; Inkeaw, et al., 2018; Dhali, et al., 2017).

Due to its writing system, segmenting Asian characters presents its own challenges and therefore needs additional techniques. In many cases of segmenting Devanagari characters, the text is commonly decomposed into lines, words and characters (Srivastav & Sahu, 2016; Mehul, et al., 2014) due to the presence of *shirorekha*, which is a straight line connecting each character in a word. In segmenting Chinese characters, Mei et. al. implemented 3 stages of segmentation (Mei, et al., 2013). In the first stage, vertical white space was used as a delimiter of any character. Then, two or more segmented characters would be merged if it is identified to have a connection region, which is recognized by means of a vertical projection. The last stage is fine-gained segmentation, in which the under-segmentation cases are splitted based on the defined rules (Mei, et al., 2013).

To segment Javanese characters, Widiarti et al. applied the projection profile for line segmentation and Moving Average Algorithm to refine the vertical projection (Widiarti, et al., 2014). Meanwhile, Budhi and Adipranata in (Budhi & Adipranata, 2015) made use of attributes of a character image and skeletonizing to segment Javanese characters. Having object of Baliness characters written on palm leaf, Kesiman et al. divided their segmentation system into 4 subtasks. The first subtask is brushing character area of gray level images with minimum filtering to overcome the problem of semi-sapce areas (Kesiman, et al., 2016). The second subtask is to apply the average block projection profile, followed by the selection of the candidate area for segmentation path. The fourth subtask deals with the construction of non-linear segmentation path by implementing the multistage graph search algorithm (Kesiman, et al., 2016).

3 METHODS

The proposed algorithm consists of two main steps i.e. the background removal and the text segmentation. The background removal was performed through image binarisation. We simply applied the binarisation function provided in Python and used threshold t=160 for the current data samples. This process is to separate the necessary pixels forming the foreground part – the text which is commonly marked as 1 -- from its background (marked as 0). Following it is the so-called segmentation process which was applied to the foreground pixels only.

The segmentation process was designed to comprise three stages of line segmentation and a stage of character segmentation. Such segmentation model is aimed to adjust the nature of the manuscript image and the Javanese script to the Projection-Profile Cutting (PPC). From this point forward, we would like to use the term ‘pass’ to refer to a ‘stage’ in our line segmentation.

3.1 Vertical Pass 1: Finding the Line Candidates

The first pass is to find the possible text line candidates from the binarised image. PPC is used here by calculating the binary vertical histogram of horizontal foreground pixel projection along the image height and selecting vertical (y) values which is changing from 0 to > 0 as a line start point and from > 0 to 0 as a line end point. These pairs of ordinates are then saved as an array as the first most possible candidates of line. Illustration of Pass 1 result applied to original image is shown in Figure 1.

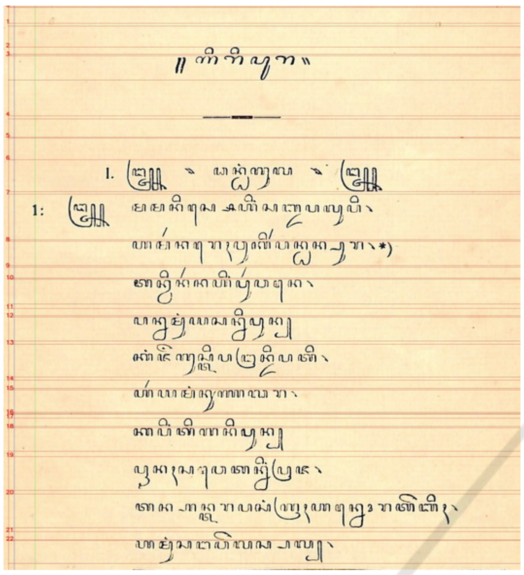
Figure 1: Illustration of Pass 1 result for test image dn0027.jpg, 23 line candidates detected.

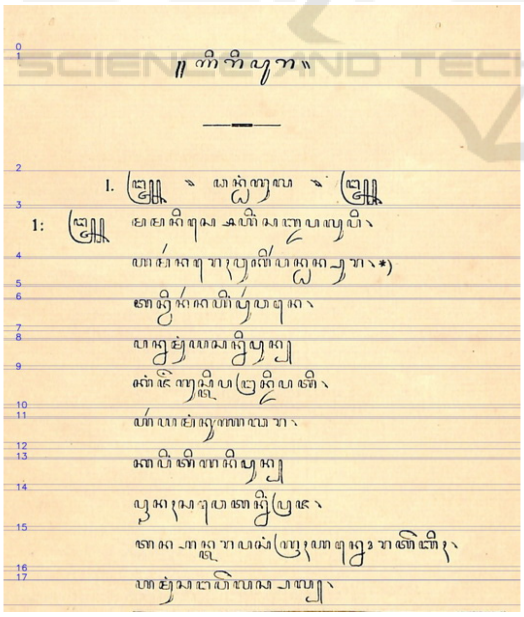
Figure 2: Illustration of Pass 2 result for test image dn0027.jpg, 18 true line candidates left, noise lines removed.

3.2 Vertical Pass 2: Removing False Lines

The lines obtained from pass 1 are not free from false lines yet. These false lines may be formed by noises which regarded as foreground pixels by the binarisation process, and thus may be detected as a line. These false lines are having a very small height value, so all lines having a height value less than 5 pixels are removed from the array. This may not eliminate all false lines, but at least it does minimise the possibility of having a false line and reduce the errors in calculating the average line height which is to be done in the next pass. The illustration of Pass 2 result applied to original image is shown in Figure 2.

3.3 Vertical Pass 3: Refining, Splitting and Merging

The first purpose of this pass is to ensure that the remaining lines in the array are truly text lines. To achieve this we performed statistical analysis to the vertical histogram and calculates the First Quartile (Q_1) of the data. Then we re-examine the histogram for hills between zeros (which represents a text line) and check whether there are ordinate (y) members of the hill which exceeds the Q_1. The pair of start and end ordinate of these values over Q_1 are then recorded as true line candidates.

These true line candidates then compared against array of lines obtained from Pass 2. Lines that are containing at least one true ordinate pairs are kept in the array. From calculating these true line candidate pairs we also found that the average height value between start and end ordinates are an approximation of the original plain character height h_O, the height of a Javanese *carakan*, a character without either *pasangan* (pseudo-consonant cluster maker) or *sandhangan* (diacritics) for the document.

This knowledge is then used for the merging process, in which true line candidates with less height than h_O are possibly lines of *sandhangan* or *pasangan* which should be merged to a text line, either from above or from below. This is done by calculating the vertical space between an allegedly *sandhangan* or *pasangan* line with nearby text lines. If the vertical space is less than 20 percent of h_O, the sandhangan or pasangan line is then merged with the text line.

The last step performed in this pass is splitting, in which if a text line in the array is found to have height more than 3 times of h_O, it is analysed and the integer result of its height divided by $3 \times h_O$ are used to split its entire height into 3 lines of equal height. This method is not always successful and sometimes it results in over segmentation.

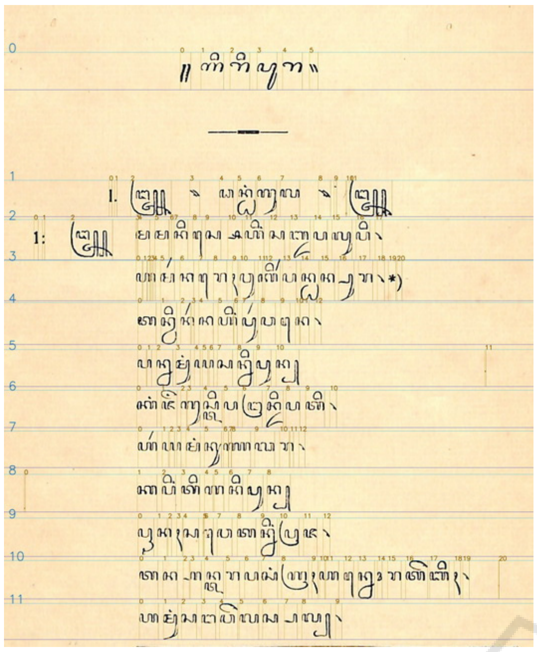

Figure 3: Illustration of Pass 3 result for test image dn0027.jpg (blue) and Horizontal Pass (brown). 12 true text lines left after reassuring, splitting and merging procedure applied.

3.4 Horizontal Pass

After the first 3 vertical pass, it is assumed that the text lines stored in the line array are the true ones, and so for each line PPC is used horizontally to segment the characters inside the line. A histogram is calculated for the vertical projection of each line, and collecting horizontal (x) values which is changing from 0 to > 0 as a character start point and from > 0 to 0 as a character end point. These pair of x values are then stored as character array for each line. Illustration of Pass 3 and Horizontal Pass results up to this point is shown in Figure 3, figuratively applied to original image.

The character segments are then formed by cutting along the line and character boundaries obtained from the stored arrays, trim the remaining vertical blank space, and save the segment as separate images in the server. However, these four efforts are not been able to eliminate noises occurring within a true line, since there is no special function created to check and eradicate such noise in the Horizontal Pass, due to save the execution time. The final result of segmented characters is illustrated in Figure 4.

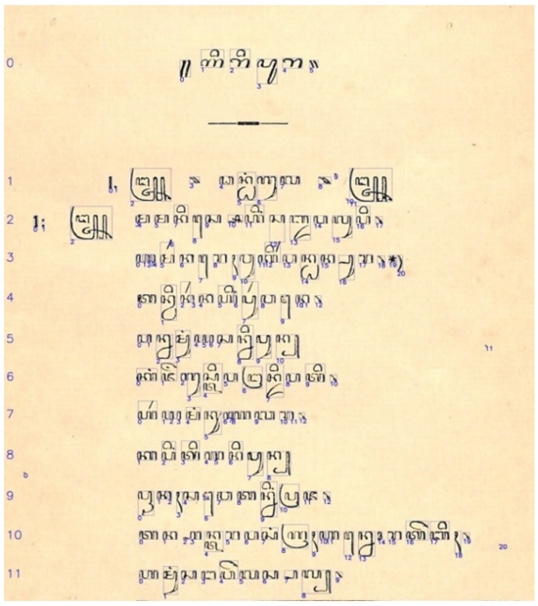

Figure 4: Illustration of final character segments, numbered from left for each text lines.

4 EXPERIMENT AND RESULTS

Seventy two Javanese document images of 24-bit JPG format were used for the character segmentation experiment; each page should roughly contain between 150 to 300 character segments. These samples are produced by randomly scanning pages from Serat Mangkunegara IV, Wulang Maca, and Kitab Rum manuscripts using a high-resolution document scanner, the Fujitsu ScanSnap SV600 with a resolution not less than 400dpi. For testing purpose, the authors wrote a program according to the proposed algorithm using Python 3.7 and MySQL database to store the results.

4.1 Line Segmentation

For the assessment scheme, we conducted two levels of evaluation: the line and character segmentations. This scheme is aimed to monitor how far the proposed method affects the accuracy of the line segmentation and to minimize the trickle-down effect of line segmentation error on the character segmentation. Furthermore, we are able to identify and cope with the problems in case the character segmentation rate is unsatisfactory.

14 out of 72 pages were randomly chosen as sample for evaluating the line segmentation. 2 out of these 14 pages contain several skewed lines as a result of poor scanning quality. The other 2 pages have very

narrow spaces among their lines; the character size is very small and dense. This sample set produces in total 268 lines. We considered these lines as the ground truth data. The information on the document sample IDs, the ground truth line numbers and the number of lines segmented by the system on the Vertical Pass 2 and 3 could be found on Table 1.

Table 1: The number of lines in the test documents as ground truth and the number of lines segmented in the vertical Pass 2 and 3

TestdocID	Pass 2	Pass 3	Ground truth
d006	30	21	21
d009	26	21	21
d010	30	21	21
d015	33	21	21
d017	31	23	21
d019	27	23	21
d027	32	21	21
dn031	28	21	21
inj525	18	18	18
inj529	27	27	27
wm003	10	10	10
wm008	24	18	16
wm1009023	14	11	10
zkb01	34	19	19

The evaluation on line segmentation was conducted on the outputs of Vertical pass 2 and 3. The performance of vertical pass 1 was left unassessed since it produced the segment candidates. The Mean Absolute Error (MAE) metric was selected to forecast the accuracy. The reason is that it is much more effortless to mark the line segmentation error than to examine and count the true ones. The MAE rate is computed based on the Equation 1, where n refers to the number of examined line samples, $O_{i,ref}$ refers to the number of text lines in a referent object, and $O_{i,est}$ is the number of lines obtained in the texts by the applied method.

$$MAE= \frac{1}{n} \sum_{i=1}^{n} \left(O_{i,ref} - O_{i,est}\right) \qquad [1]$$

One drawback of MAE is that it ignores the type of segmentation error. For this reason, we distinguished the line segmentation error in two types, i.e. under-segmentation (US) and over-segmentation (OS). The US is identified when two lines of characters are left being unsegmented by the system and OS when a line is segmented into 2 or more lines. Table 2 shows that the proposed methods produce more OS errors than US. This Table shows also that the refinement of PPC has successfully reduced the error rate to 0.32. In other words, it increases the line segmentation accuracy by ca. 32.8%.

Table 2: The number of over-segmented lines (OS), under-segmented lines (US) and the Mean Absolute Error.

Passes	#OS	#US	MAE
Pass 2	91	4	0.354
Pass 3	7	0	0.026

4.2 Character Segmentation

The segmentation tests cost roughly 2-4 minutes per document image, and the output of the program also present an easily identifiable numbered line and character segments as shown in Figure 4, for human evaluation purpose. The proposed algorithm results in 19,112 segments for the whole document images in test. To evaluate the system outputs, we collaborated with Javanese Wikipedia Community (Komunitas Wikipedia Jawa) which verified them manually in 4 batches. Only 15.386 segments were verified, in which 73.59% (11.322) system outputs are correctly segmented, 22.5% (3.464) are over-segmented, 1.3% (206) are under-segmented, and the rest has not been labelled as either one of three categories above. The incorrect segmentation causes are mentioned below.

Over-segmentation occurs when a supposedly single segment is segmented as more than one. This is almost always happened during the character (horizontal) segmentation. It is due to the characters having not been equally inked when printing so when the binarisation is carried on, some of the linkage in the symbol is broken and causing horizontal blank spaces within a character, as shown in Figure 5. At experiments we were trying to lower the threshold *t* by tens to *t=150* and *t=140* in order to gain more foreground pixels in between but this does not help significantly.

In some rare case, over-segmentation also occurs vertically. It might have caused by line height identification failure during the Vertical Pass 3.

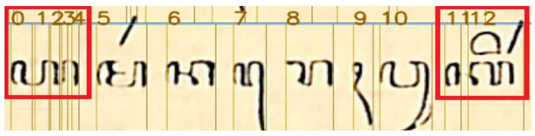

Figure 5: Example of over segmentation caused by broken linkage inside characters

In contrast with over-segmentation, under-segmentation is occurred when a supposedly few different segments are regarded as only one. The only cause of this is the lack of vertical or horizontal straight blank space between lines or characters, as demanded by PPC. Under-segmentation both occurred in lines or characters level and the Vertical Pass 3 sometimes cannot cope with it as the average forced vertical segmentation may cut in the wrong place.

Also, due to the natural condition of the manuscript, some noises are just at the right size and intensity or having similar traits to a character to get through all four segmentation stages, thus survived to be regarded as a valid line or character. Our proposed algorithm is not designed to detect these kinds of noises, and so it has to be evaluated manually to dispose of the incorrect segments.

5 CONCLUSIONS

The proposed algorithm is capable of improving the original Projection Profile Cutting (PPC) for line segmentation in printed Javanese manuscripts. After Vertical Pass 3 the number of text line segment which is equal to ground truth is 10 from 14 test documents, which is 71.42% correct. It is increased from 2 test documents (14.28%) after Vertical Pass 2, and 0 documents from Vertical Pass 1 which applied straight PPC. It also reduces over-segmentation and under-segmentation of lines by having MAE of 0.32, which in other words, increases the line segmentation accuracy by ca. 32.8%. An improved text line segmentation is crucial to provide a good base for character segmentation.

Since PPC is based on straight vertical and horizontal blank spaces, without further refinements it would be incapable of segmenting Javanese characters to its individual units, thus the resulting segments are mixed of individual symbols and syllables. This may be adequate for designing an Optical Character Recognition (OCR) system, but may cause problems when used in designing a transliteration or translation system.

ACKNOWLEDGEMENTS

We would like to give our deepest gratitude to Wikimedia Indonesia which has funded this research and our participation in this conference. We also give our thanks to Javanese Wikipedia community for their contribution on evaluating the character segments.

REFERENCES

Asif, M. et al., 2017. Multinational vehicle license plate detection in complex backgrounds. *Journal of Visual Communication and Image Representation,* 46(C), pp. 176-186.

Budhi, G. & Adipranata, R., 2015. Handwritten Javanese Characters Recognition Using Several Artificial Neural Network Methods. *Journal of ICT Research and Applications,* 8(3), pp. 195-212.

Chaiwatanaphana, S., Pluempitiwiriyawej, C. & Wangsiripitak, S., 2017. Printed Thai Character Recognition Using Shape Classification in Video Sequence Along a Line. *Engineering Journal,,* 21(6), pp. 37-45.

Dhali, M. et al., 2017. *A Digital Palaeographic Approach towards Writer Identification in the Dead Sea Scrolls.* s.l., s.n., pp. 693-702.

Ding, C., Utiyama, M. & Sumita, E., 2018. *Simplified Abugidas.* Melbourne, Australia, ACL, pp. 491-495.

Inkeaw, P. et al., 2018. Recognition-based Character Segmentation for Multi-level Writing Style. *International Journal on Document Analysis and Recognition,* 21(1-2), pp. 21-39.

Karthikeyan, K., Vijayalakshmi, V. & Jeyakumar, 2013. License Plate Segmentation Using Connected Component Analysis. *OSR Journal of Electronics and Communication Engineering (IOSR-JECE),* 4(5), pp. 18-24.

Kesiman, M., Burie, J. & Ogier, J., 2016. *A New Scheme for TextLine and Character Segmentation from Gray Scale Images of Palm Leaf Manuscript.* Shenzhen,China, s.n., pp. 325-330,.

Lacerda, E. & Mello, C., 2013. Segmentation of Connected Handwritten Digits Using Self-Organizing Maps. *International Journal of Expert System with Application,* 40(15), pp. 5867-5877.

Lue, H. et al., 2010. A Novel Character Segmentation Method for Text Images Captured by Cameras. *ETRI Journal,* 32(5), pp. 729-739.

Manmatha, R. & Rath, T. M., 2003. *Indexing of Handwritten Historical Documents - Recent Progress,* s.l.: Semantic Scholar.

Mehul, G. et al., 2014. *ext-Based Image Segmentation Methodology.* s.l., Elsevier, Procedia Technology.

Mei, Y., Wang, X. & Wang, J., 2013. *An Efficient Character Segmentation Algorithm for Printed Chinese Documents.* s.l., s.n., pp. 183-189.

Srivastav, A. & Sahu, N., 2016. Segmentation of Devanagari Handwritten Characters. *International Journal of Computer Applications,* 142(14), pp. 15-18.