

Preprocessing Model of Manuscripts in Javanese Characters

Anastasia Rita Widiarti¹, Agus Harjoko², Marsono³, Sri Hartati²

¹Department of Informatics Engineering, Sanata Dharma University, Yogyakarta, Indonesia

²Department of Computer Science and Electronics, Gadjah Mada University, Yogyakarta, Indonesia

³Department of Nusantara Literature, Gadjah Mada University, Yogyakarta, Indonesia

Email: rita_widiarti@yahoo.com, aharjoko@ugm.ac.id, marsono@ugm.ac.id, shartati@ugm.ac.id

Received 12 August 2014; revised 10 September 2014; accepted 5 October 2014

Copyright © 2014 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Manuscript preprocessing is the earliest stage in transliteration process of manuscripts in Javanese scripts. Manuscript preprocessing stage is aimed to produce images of letters which form the manuscripts to be processed further in manuscript transliteration system. There are four main steps in manuscript preprocessing, which are manuscript binarization, noise reduction, line segmentation, and character segmentation for every line image produced by line segmentation. The result of the test on parts of PB.A57 manuscript which contains 291 character images, with 95% level of confidence concluded that the success percentage of preprocessing in producing Javanese character images ranged 85.9% - 94.82%.

Keywords

Binarization, Characters Segmentation, Line Segmentation, Noise Reduction, Manuscript Image Preprocessing

1. Introduction

Manuscripts in Javanese characters are an asset of Javanese culture which must be preserved. One of the ways to preserve manuscripts is digitizing the manuscripts by taking pictures or scanning. Manuscript images from manuscript digitations are then stored correctly, safely and in media which aren't easily destroyed.

Results of manuscript digitization can be digital assets as well as for research interests, such as to explore important and useful information in those manuscripts. However, one of the emerging issues is the manuscripts are written in Javanese characters, which not many people are able to read the manuscripts. Manuscript transliteration from Javanese characters to Roman characters manually as well as automatically is one of the methods for Javanese people to use valuable information from the manuscripts.

Manuscript preprocessing is the first and very vital part of manuscript automatic transliteration. The success of manuscript transliteration determines the success of the process of converting characters from Javanese characters into Roman characters. The final result of manuscript preprocessing process is a series of images of Javanese characters which forms the manuscript image which is the input data.

Preprocessing manuscripts in Javanese characters is often not easy. The difficulty in preprocessing is caused by various things, *i.e.*:

- Original condition of the manuscripts is not clean. This will make images of digitations result no clear, so colors which can differentiate objects from backgrounds is sometimes unclear.
- Manuscript digitations process is imperfect, for example due to low lighting so that manuscript images aren't clear, manuscript positioning in digitization is not maximal because the manuscripts can't be opened widely, which makes digitations results tilted or look rolled.
- Original characteristics of character images in the manuscripts due to writing methods of characters in that period.

Based on various studies it's discovered that most original manuscript condition is fragile, whether due to the material of the paper used, imperfect manuscript storing process, non-white or yellowish manuscript paper color which looks like there are dark spots, so that images of characters in manuscripts are unclear. These manuscript conditions cause input for studies can only be collected from copies of manuscripts from museum staffs, not from photographing or scanning the manuscripts directly. This step is expected to not make manuscripts more fragile, but makes input data inauthentic.

Based on digitization of copies of manuscripts using a scanner with 300 dpi capacity, it's discovered that in several parts of manuscript digitization results there are slight difference in color gradation between object and background, and there are noise objects due to previous photocopying process of the manuscripts. This will raise new problem in manuscript preprocessing, because there are characters which look like objects but are considered background, and vice versa.

The two paragraphs above explain challenges in segmentation due to original condition of input images or physical condition of input images. Another challenge is due to characteristics of characters in the manuscripts, which are:

- Limits between lines in manuscripts aren't clear. Unclear limit between lines is caused by two characters or more in different lines which are in the area of another line, or even connected with each other. For example, **Figure 1** shows two images of Javanese characters *mmi* and *ge* in two different lines which have intersecting parts of characters in an area, *i.e.* part of *sandangan pepet* character of Javanese character *ge* is in the area of the line above it.
- Limits between characters in a line aren't clear. Unclear limit between characters can be caused by unclear limit between lines or part of a line overlapping part of another line, there are two or more interconnected characters, or they way the characters are written. Sample part of PB.A57 manuscript in **Figure 2** shows two different lines which contain parts of another line. This is because the writer's writing style is tilted, making the next character enter the area of the previous character.

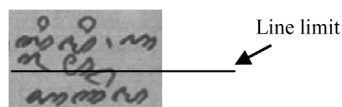


Figure 1. Example of two different lines of a part of PB.A57 manuscript which contains two different characters with intersecting positions.

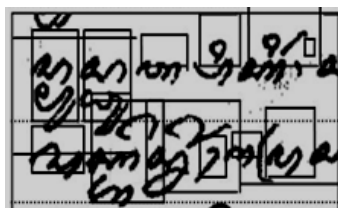


Figure 2. Sample of part of PB.A57 manuscript which contains unclear limit between characters.

- A character may contain only one object or a collection of two or three objects. For example an image from PB.A57 manuscript presented in **Figure 3** show character *ha* which contains one object, character pasangan *pi* or *_pi* which contains two objects which are *_pa* under it and *wulu* above it, and character *kir* which contains three objects which are character *ka* under it, *sandangan wulu* in the upper left, and *layar* in the upper right, respectively.

2. Literature Review

Tangwongsan, and Sumetphong [1] develop a character recognition system to recognizes images of damaged historic documents from Thailand. There are three main stages in the developed system, which are data preparation process, segmentation stage, and character recognition stage. In data preparation stage there are two main processes, which are document image binarization using Otsu method, and fixing the slope of the document using Hough's transformation method which has been developed by Amin and Fischer. In segmentation stage two methods are applied, *i.e.* profile projection and object cropping method developed by Pal, and Datta. Surinta, and Chamchong [2] also successfully perform segmentation on historic image written by hand on palm leaves. It starts with binarization stage using Otsu method to separate objects from backgrounds, followed by line segmentation stage by applying profile projection method, and last is segmentation stage to obtain characters by using histograms of segmented images. Considering two studies above, manuscript preprocessing stage will consist of binarization, fixing slope, then segmentation stage.

Casey and Lecolinet [3] state that one of the basic strategies of character image segmentation, which is segmentation, is performed based on the characteristics of the character which will be segmented, for example height of the character, width of the character, and distance with neighboring points. Garg, *et al.* [4] perform line, word, and character segmentations of Hindi scripts using header line and baseline detection approaches which have been adjusted with the characteristics of Hindi manuscripts. This Hindi segmentation is a complex problem because words are a group of connected characters and form *shirorekha*. Furthermore, sometimes conjuncts happen. By using knowledge of characteristics of Hindi manuscripts, Garg discovers initializations of minimum height of consonants, average character height, and requires text slope to not be higher than the height of consonant characters. Lehal and Singh [5] study segmentation method on Gurmukhi text using a combination statistical analysis of the text, profile projection, and analysis of connected components. By using horizontal projection they discover segmentation failures which are divided into two failures, which are over-segmentation and under-segmentation. Over segmentation happens because there is distance between lines of the text, and under segmentation happens because there are parts of the text that overlap other lines. Lehal and Singh then determine several unit values based on characteristics of character positioning method, such as height of an area, determination of strip classes, which are character groups in a line, and determination of height of the first line. Palakollu, *et al.* [6] also study line segmentation method on Hindi manuscript images using projection-based approach. They build an algorithm to detect header line and baseline based on several initial assumptions such as average line height which is 30 pixels, to estimate real average line height. The four studies related to character segmentation above inspire this solution for problems of segmentation of manuscripts in Javanese characters which have many segmentation challenges.

3. Proposed Method

Replacing images of Javanese characters from a manuscript into Roman characters is influenced by the quality of input character images, so the stage of preparing character images is very important. Manuscript preprocessing to produce images of characters which form a manuscript is also aimed to improve the quality of the character images produced, such as to clear the difference between objects and backgrounds and removing noises which will disturb transliteration.



Figure 3. Sample of Javanese characters which consist of one, two, and three objects, from left to right are Javanese character *ha*, *_pi*, and *kir* from PB.A5 manuscript.

Input preprocessing is producing manuscript image from scanning using a scanner on a copy of a manuscript. This is performed to preserve the manuscript because scanning by a scanner or camera directly will damage the original script. **Figure 4** shows a flow chart of the manuscript preprocessing which contains four main steps.

The first step in preprocessing was image binarization process to turn input images into binary image format, which is image in black or white, or 0 or 1. Not only because color information was no longer needed in the next process, new data format would reduce the complexity of computation, because data format which was processed was only in 0 or 1 format. Based on the study of Widiarti and Harjoko [7], the method used for binarization was Otsu method because the method was proven to be able to be used in the binarization process of images of printed text documents of manuscripts in Javanese characters.

Observation on several new images from binarization showed that there were noises which were mostly collection of spots like sand grain in a rather large number, so the next step was noise reduction. Objects which were considered noises were object with height and width between 1 pixel and 7 pixels. These values were obtained from the conclusion of a study that a character object had minimum height or width 8 pixels, so if there was any object with height or width below 8, it was certainly a noise.

Simple method to get information on height and width of an object in a manuscript was following every black pixel which formed objects in manuscripts, and then giving the same label for every interconnected pixel which formed an object. Connected component operation had a significant role in helping to discover which objects must be marked with the same label. Object height was obtained from subtracting the value of the lowest line with the highest position where the object was located. Object width was obtained from subtracting rightmost column with leftmost column where the object was located.

After manuscript images are relatively free from noises, the process to obtain images of characters in the manuscript can start. Manuscript segmentation starts by finding lines of characters which form the manuscript, then followed by finding characters which form lines of characters produced by line segmentation.

To perform rough line segmentation, the first stage was "Projection stage", which was projecting object pixels vertically to obtain information about which pixel lines had pixel groups that form objects. Formulation (1) was applied to discover vertical projection result in the shape of P_v vector, size R , from binary image S with size (R, C) [8].

$$P_v [i] = \sum_{j=1}^C S [i, j]. \quad (1)$$

Sample in **Figure 5** shows the curve resulted from searching for the values of P_v vector, for S manuscript image in **Figure 6**.

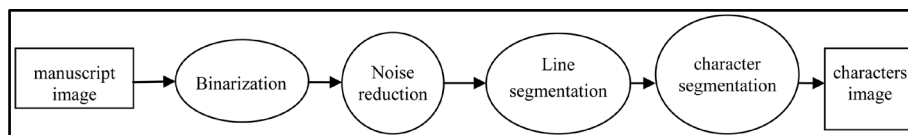


Figure 4. Flow chart of the manuscript preprocessing.

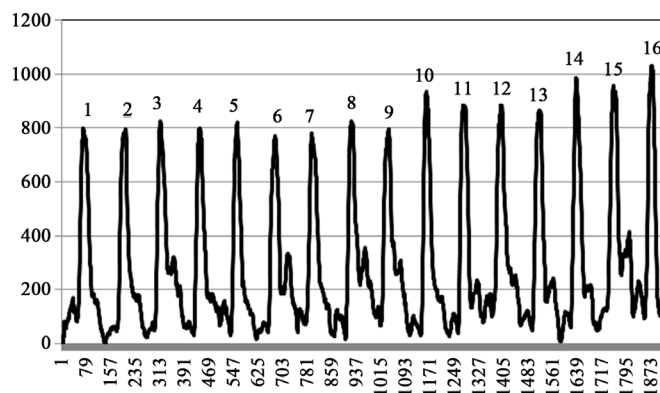


Figure 5. Curve of vertical projection of manuscript image in **Figure 6**.

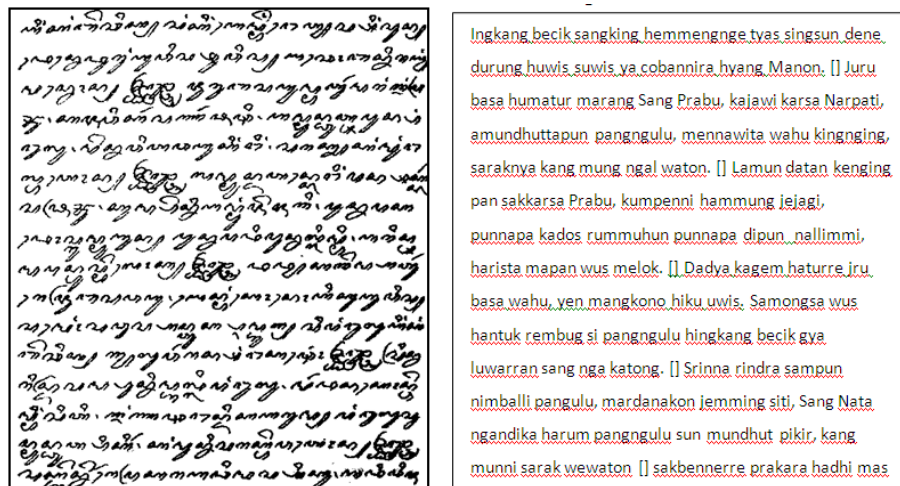


Figure 6. Part of image of manuscript in Javanese character from Sonobudoyo museum collection with catalogue number PB A.57 (P 190 Suluk Rasa Sejati PB A.57 175 Bhs Jawa Aks Jawa Macapat Rol 153 no. 9) [10].

Information of position of pixel lines and number of objects in the line position was curves with multiple peaks. If valley midpoint to the peak and then back to the next valley midpoint was called 1 phase, then if in the manuscript there were 16 lines of characters, there would be 16 phases.

Because most manuscript characteristics had overlapping image lines, curve of vertical projection should be defined to get a curve which reflected clear distance between phases. Phases in a curve with clear distance between them gave clues of the locations of the beginning and end of a line because 1 phase showed 1 line of manuscript image. One of the methods of curve refining which could be used was moving average algorithm. If there is N data $[y_1, y_2, \dots, y_N]$, by using formulation (2) new data row (Dk)s would be discovered as a result of data refining from data around it, with data use range or filter width around $2N + 1$ [9].

$$(y_k)_s = \sum_{i=-N}^{i=N} y_{k+i} / (2N + 1) \tag{2}$$

Valleys of the results of curve refining often didn't refer to the number 0 which showed that there was a gap between line because in manuscripts there were many parts where characters in the line above and below were in the same area, or intersect, or even connect. One of the ways to solve this was by determining which line had significant data shift, and that line became the reference to separate the lines. The method to determine line shift was by marking the line where was a shift of the result of subtraction of initial value with new value which had been refined from negative to positive or vice versa. Results of the test on line shift information were initial clues that in those numbers there were line shifts.

Because characteristics of manuscripts in Javanese characters often disables clear distance between lines, cropping line images using information from refined results of vertical projection often cropped several characters. For every character which was really in the range of the discovered lines, the character must be right in the line position. The problem was parts of characters in the current line often entered the area of the next line. Because of this if there was strict cropping of character line in the determined positions, several characters would be cut. To solve this, the concept of connectedness between pixels in a character was established to discover parts of characters which might be cut due to line cropping process.

It was very likely that there was a situation where the results of line cropping showed two or more different character lines which were marked as the same line or considered as one line because they're pressed closely together, or because characters in different lines connected with each other. Therefore, there should be further investigation on appropriate height of character line. **Figure 7** shows a writing of Javanese characters with limits of writing area. Understanding area limits of Javanese characters writing would be the guide for calculating appropriate character line height.

The main area to place Javanese characters is area or zone 2, which is area for the placement of *Legena* character group. While zone 1 and 3 are areas for complementary properties of *Legena* characters called *sandangan*

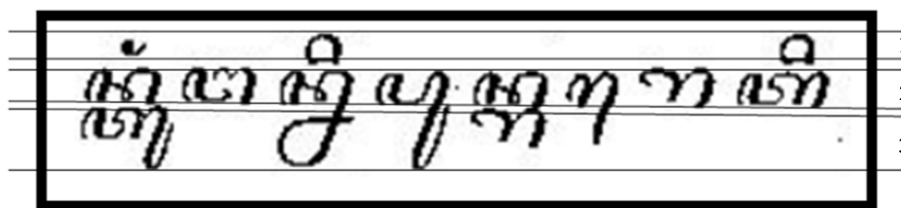


Figure 7. Methodology for writing Javanese characters in one character line.

and *pasangan*. Based on the thought that most characters of a manuscript are *Legena* characters and are in zone 2, average height of all objects which formed the manuscript could be calculated. It could be assumed that average height of the objects was the height of objects which formed main characters, which were *Legena* characters.

However, sometimes there were *sandangan* and *pasangan* along with the *Legena* characters. Height characteristic of objects which formed *sandangan* and *pasangan* was usually shorter than the height of objects which formed *Legena* character. With these assumptions, the height of objects which formed *sandangan* and *pasangan* it was approached as the standard deviation value of average height, so the height of zone 1 and 3, which were the locations of *sandangan* and *pasangan*, was the value of standard deviation of average height of all objects which formed the manuscript. Using the assumptions above, normal height of character line T_w wouldn't be bigger than average object height A_T plus two times the value of standard deviation STD_T (Formulation (3)).

$$T_w = A_T + 2 \times STD_T \quad (3)$$

By using information on the height of a character line above, it's easy to know whether a character in a line in an image produced had normal line height or not. Abnormal object height would occur when the object height was bigger than average height of normal objects plus twice standard deviation value of average height. If there was any abnormality in the height of the object, the object in the line must be two different objects from two different lines which were connected. Principally, the object might be a part of a character in another line, so it should be cropped or separated.

The next stage was character segmentation on line images produced in line segmentation stage. Figure 8(3) shows an illustration of the result of character segmentation, which were images of characters from line image in Figure 8(1).

Sample of line image in Figure 8(1) shows that there was clear distance between characters so it seemed to be possible to apply horizontal projection on the character line image to get information on which positions were possible for character cropping. Formulation (4) could be applied to discover horizontal projection result, *i.e.* vector P_h sized C , from binary image S sized (R, C) [8].

$$P_h[j] = \sum_{i=1}^R S[i, j]. \quad (4)$$

Figure 8(2) shows curve of vector Ph which was produced using the data of line image in Figure 8(1). The position information in the curve to guide the determination of character transition position hadn't produced correct character images. Mistakes in cropping position happened because character writing style is tilted, so if horizontal projection was applied directly, there were some parts of a character which merged.

To clear mistakes due to tilted writing, Figure 8(2) presents a line curve from horizontal projection of Figure 8(1). From the results of vertical projection, several turning points which became indicators that there was character transition at that point. Of sixteen characters which should be found in Figure 8(1), only 4 characters were segmented correctly, *i.e.* in Figure 8(3)e, Figure 8(3)i, Figure 8(3)j, and Figure 8(3)k. The mistake happened because there was one character which became two characters, for example in Figure 8(3)a, there were parts of the character which became parts of Figure 8(3)b, Figure 8(3)g, and Figure 8(3)h should also be one character group. Another mistake happened because several characters were segmented as one character as shown in Figure 8(3)d which was three characters being one character group.

Mistakes due to wrong character segmentation above could be solved by first changing the tilt of line image into perpendicular to vertical axis. With the tilt change, horizontal projection was expected to produce optimum clue to discover the positions of character cropping. The illustration in Figure 9(1) shows image change in Figure 8(1) due to tilt change operation, and new curve from the horizontal projection (Figure 9(2)) and illustration

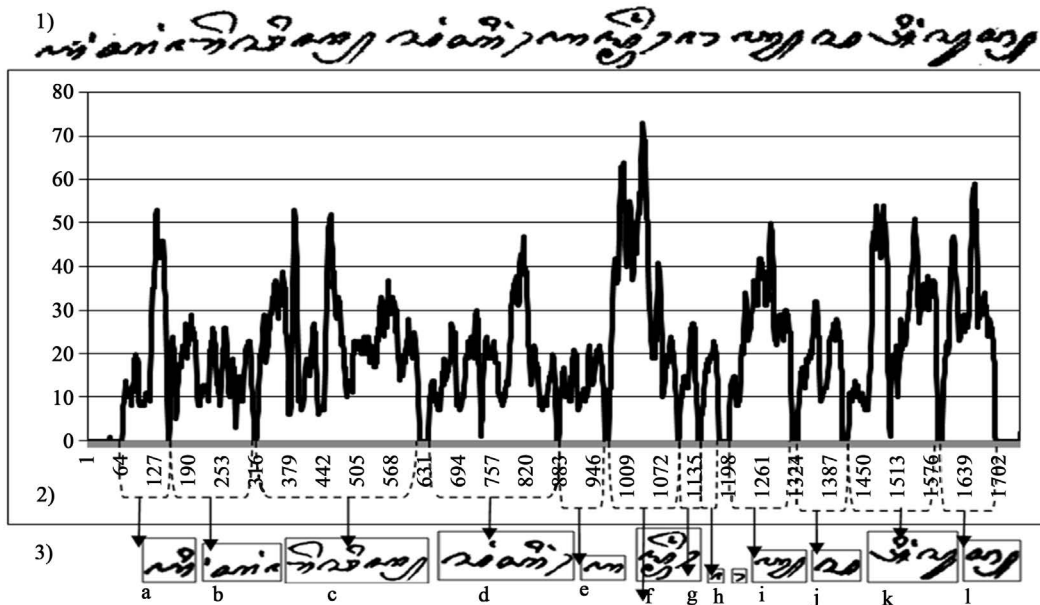


Figure 8. Sample of line image along with curve from vertical projection and the character mapping based on the curve.

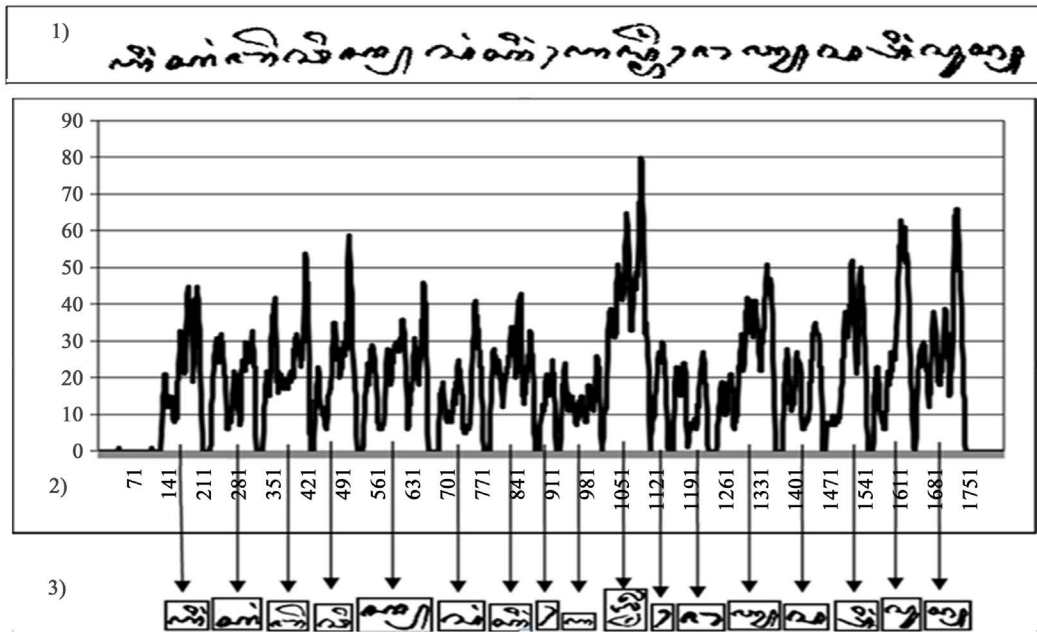


Figure 9. Image in Figure 8(1) which has been straightened and curve from horizontal projection and result of character image cropping.

of the result of character image cropping using information in the new curve (Figure 9(3)).

Based on the descriptions of initial problems of character segmentation from line segmentation, the first step in character segmentation was tilt repair stage. In several cases after the tilt of images were repaired, character images were produced optimally, but in other cases repairing tilt didn't always produce horizontal projection curve which strictly contained position information of separation of character images. Unclear distance between characters happened because a character written was in the area of another character or even connected with it. In case the result of character segmentation still considered groups of two or more characters as one character, character segmentation results should be processed further.

Sample image from part of image of manuscript PB.A57 in the lower part of **Figure 10** marked by an arrow is sample of wrong result of character segmentation. The mistake happened because there were 2 character images which were segmented into 1 character, although in the original image as shown in the upper part of **Figure 10**, the tilt had been repaired as seen in the middle part of **Figure 10**.

In cases such as in **Figure 10**, there should be further repair, in this case cropping again the results of character segmentation which still contained 2 or more character images. To produce a series of characters which had been separated well, there should be investigation on whether character width was normal or not. Character width was normal if the width wasn't bigger than average width of all objects which formed the manuscript plus standard deviation. If character width produced was bigger than average width of objects in the manuscript plus standard deviation of object width, the object was almost certainly two or more characters considered as one character. In characters which were indicated to have abnormal width, further process to separate the characters using connectedness operation between characters which formed character objects was required.

4. Execution Result

Data of the main test used for model tested was taken from manuscript with catalogue number PB.A57 as shown in **Figure 6**. Using the data of manuscript PB.A57 which had gone through binarization and noise reduction processes would produce line images from the results of line segmentation as shown in **Figure 11**.

Image in line 1 **Figure 11** shows very clean result of line segmentation, meaning the image was segmented well without any connection with another line. This happened because there is clear distance between the first and second lines in the original image. Meanwhile, starting from the image of line one and so on, there are parts of character images in different lines which are in the same area or are connected. By using information on average object height and standard deviation, line separation or cropping could be performed.

After line segmentation, the next stage was character image segmentation of each line image from line segmentation process. **Table 1** shows a sample of character segmentation result of the first line image of manuscript PB.A57, along with correct and wrong information which states whether the results of character segmentations were correct or not. A segmentation result was correct if there was no missing part of the character and only contained one character.

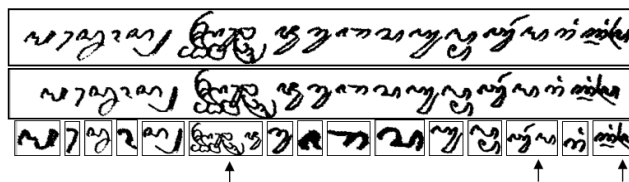


Figure 10. Part of line image from manuscript PB.A57 along with result of tilt repair of the image and result of character segmentation.

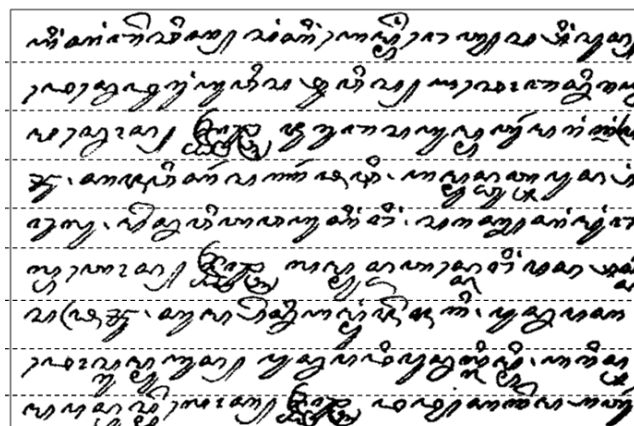






















Figure 11. Part of manuscript in **Figure 6** with lines which show line transition.

Table 1. Sample of character image segmentation in Figure 8(1).

No.	Result of Characters Segmentation		
	Character image	Correct/Incorrect	Information
1		Correct	
2		Correct	
3		Incorrect	Front part of character <i>be</i>
4		Incorrect	Back part of character <i>be</i>
5		Correct	
6		Correct	
7		Correct	
8		Correct	
9		Correct	
10		Correct	
11		Correct	
12		Correct	
13		Correct	
14		Incorrect	Front part of character <i>nga</i>
15		Incorrect	Back part of character <i>nga</i>
16		Correct	
17		Correct	
18		Correct	
19		Correct	
20		Correct	

5. Result Analysis

To obtain data on the range of percentage of truth of segmentation model test, interval proportion test was performed using processed data on information on the correctness or incorrectness of segmentation results. The number of original character images from the manuscript which was the object of model test was 291 characters. By using Slovin's formulation [11] in Formulation (5), then minimum number of sample L required for statistical test with population $Q = 291$ as 168. Variable e^2 stated the percentage of inaccuracy tolerance.

$$L = \frac{Q}{1 + (Qe^2)} \tag{5}$$

Assuming the number of samples was larger than 30 and that population was unlimited, Formulation (6) is used for interval proportion test. If data of calculation of percentage of truth h from the data is L , and by using level of confidence Z , then range of percentage of truth of a test V was as shown in Formulation (6) [12].

$$h - Z_{\alpha/2} \times \sqrt{\frac{h(1-h)}{L}} < V < h + Z_{\alpha/2} \times \sqrt{\frac{h(1-h)}{L}} \tag{6}$$

The next step in interval proportion test was to discover 168 results of character segmentation randomly to check whether the results of character segmentation were correct or not. Every time a character image in the manuscript was, taken, it would be checked whether the result of character segmentation of the character was

segmented correctly or not. Correct segmentations were given score 1, while wrong ones were given 0, to ease calculation the number of correct character segments.

Using the method above, the number of total correct character segments could be calculated. As an example for collecting 168 random characters during the first collection, there was information that there were 153 correctly segmented character images, so percentage of truth of segmentation for the first test was 91.07%.

To get accurate results, testing series above was performed 4 more times. During the fourth test, percentages of truth of segmentation were 89.88%, 92.86%, 89.286%, and 88.69% respectively. From five data of percentage of truth of segmentation, average percentage of truth was 90.36%. By using Formulation (6) and using average percentage of truth of transliteration 90.36% as h , level of confidence Z 95%, sum of characters L 168, confidence interval of average percentage of truth of segmentation V in the manuscript was:

$$\Leftrightarrow 90.36\% - 1.96 \times \sqrt{\frac{90.36\% (1 - 90.36\%)}{168}} < V < 90.36\% + 1.96 \times \sqrt{\frac{90.36\% (1 - 90.36\%)}{168}}$$

$$\Leftrightarrow 85.9\% < V < 94.82\%$$

From the result of interval proportion test above, it was interpreted that 95% could be trusted, that average percentage of truth of all result of manuscript segmentation using the proposed character segmentation model wasn't less than 85.9%, but not larger than 94.82%.

6. Conclusion

The manuscript preprocessing method presented in this paper included methods for image binarization, noise reduction, and character segmentation in the manuscript, without removing important and main information of images of the character in the manuscript. Preprocessing products which were character images could be processed further for manuscript transliteration. The results of transliteration test of character images produced in preprocessing stage on manuscript PB.A57, with confidence level 95%, could be concluded as percentage of success of transliteration would not be less than 73.51%, for percentage of success of preprocessing in the range of 85.9% to 94.82%. Correlation level between preprocessing success and transliteration success was 77.77%, showing that manuscript transliteration success was directly related to manuscript preprocessing success. This preprocessing method still can be developed to help manuscript transliteration process which in reality often faces damages, for example unclear paper color, unclear writing on manuscripts, parts of manuscripts eaten by termites, and parts of manuscripts missing because they're ripped.

Acknowledgements

The authors would like to acknowledge the Indonesian Directorate General of Higher Education (DIKTI) and Fund Management Institution of Education (LPDP) from Ministry of Finance Republic Indonesia for supporting this research.

References

- [1] Tangwongsan, S. and Sumetphong, C. (2008) Optical Character Recognition Techniques for Restoration of Thai Historical Documents. *Proceeding of the International Conference on Computer and Electrical Engineering*, Thailand, 20-22 December 2008, 531-535.
- [2] Surinta, O. and Chamchong, R. (2008) Image Segmentation of Historical Handwriting from Palm Leaf Manuscripts. In: Shi, Z.Z., Mercier-Laurent, E. and Leake, D., Eds., *IFIP International Federation for Information Processing, Intelligent Information Processing IV*, Vol. 288, Springer, Boston, 182-189. http://www.wbi.msu.ac.th/file/721/doc_57.pdf
- [3] Casey, R.G. and Licolinet, E. (1995) Strategies in Character Segmentation: A Survey. *Proceeding of the Third International Conference on Document Analysis and Recognition (ICDAR)*, Canada, 14-16 August 1995, 1028-1033. <http://perso.telecom-paristech.fr/~elc/papers/pami96.pdf>
- [4] Garg, N.K., Kaur, L. and Jindal, M.K. (2010) Segmentation of Handwritten Hindi Text. *International Journal of Computer Applications*, **1**, 19-23. <http://www.ijcaonline.org/journal/number4/pxc387222.pdf>
- [5] Lehal, G.S. and Singh, C. (2001) A Technique for Segmentation of Gurmukhi Text. *Computer Analysis of Images and Patterns, Lecture Notes in Computer Science*, **2124**, 191-200. <http://advancedcentrepunjabi.org/pdf/A%20technique%20for%20segmentation%20of%20gurmukhi%20text.pdf>
- [6] Palakollu, S., Dhir, R. and Rani, R. (2011) A New Technique for Line Segmentation of Handwritten Hindi Text. *Spe-*

- cial Issue of International Journal of Computer Applications (0975-8887) on Electronics, Information and Communication Engineering—ICEICE*, 5. <http://research.ijcaonline.org/iceice/number5/iceice033.pdf>
- [7] Widiarti, A.R. and Harjoko, A. (2006) Pengenalan Citra Dokumen Sastra Jawa (Studi Kasus Pada Buku Sastra Jawa: Menak Sorangan I dan Panji Sekar). *SAINS dan SIBERNETIKA*, **19**, 269-282.
- [8] Zramdini, A. and Ingold, R. (1993) Optical Font Recognition from Projection Profiles. *Electronic Publishing*, **6**, 249-260. <http://cajun.cs.nott.ac.uk/compsci/epo/papers/volume6/issue3/ofr.pdf>
- [9] Efstathiou, C.E. Signal Smoothing Algorithms. http://www.chem.uoa.gr/applets/appletsmooth/app1_smooth2.html
- [10] Behrend, T.E. (1990) Katalog Induk Naskah-Naskah Nusantara Jilid I Museum Sonobudaya Yogyakarta. Djambatan, Jakarta.
- [11] Gay, L.R. and Diehl, P.L. (1992) *Research Methods for Business and Management*. Mc. Millan Publishing Company, New York.
- [12] Wackerly, D.D., Mendenhall, W. and Scheaffer, R.L. (2008) *Mathematical Statistics with Applications*. 7th Edition, Thomson Learning, Inc., USA. http://fvela.files.wordpress.com/2012/01/mathematical_statistics_with_applications1.pdf

Scientific Research Publishing (SCIRP) is one of the largest Open Access journal publishers. It is currently publishing more than 200 open access, online, peer-reviewed journals covering a wide range of academic disciplines. SCIRP serves the worldwide academic communities and contributes to the progress and application of science with its publication.

Other selected journals from SCIRP are listed as below. Submit your manuscript to us via either submit@scirp.org or [Online Submission Portal](#).

