Kannada Text Line Extraction Based on Energy Minimization and Skew Correction

Sunanda Dixit
Information Science and Engineering Department
Dayananda Sagar College of Engineering
Bangalore, India
sunanda.bms@gmail.com

Suresh Hosahalli Narayan
Department of Instrumentation Technolgy
Bangalore Institute of Technology
Bangalore, India
hn.suresh@rediffmail.com

Mahesh Belur
Bangalore, India
mahesh_ait@hotmail.com

***Abstract*- There are many governmental, cultural, commercial and educational organizations that manage large number of manuscript textual information. Kannada being one of the official languages of South India, such organizations include Kannada handwritten documents. Text line segmentation in such documents remains an open document analysis problem. Detection and correction of skew angle of the segmented text lines become another important step in document analysis. Most of the segmentation algorithms, for skewed text lines, present in the literature today are sensitive to the degree of skew, direction of skew, and spacing between adjacent lines. In this paper, proposed method for the text line extraction and skew correction of the extracted text lines uses a new cost function, which considers the spacing between text lines and the skew of each text line is used. Precisely, the problem is formulated as an energy minimization problem so that the minimization of the cost function yields a set of text lines. Further it is required to efficiently correct baseline skew and fluctuations of these text lines. This proposed method also uses an efficient algorithm for baseline correction. It consists of normalizing the lower baseline to a horizontal line using a skating window approaches, thus, avoiding the segmentation of text lines into subparts. This approach copes with baselines which are skewed, fluctuating, or both. It differs from machine learning approaches which need manual pixel assignments to baselines. Experimental results show that this baseline correction approach highly improves performance.**

Keywords- Document analysis, skew angle, skew detection and correction, cost function, energy minimization, baseline skew and fluctuations, skating window approach.

I. INTRODUCTION

Beyond modern documents, a huge amount of archive and historical Kannada handwritten documents are still to be exploited by reading systems. Document analysis plays a significant role in extracting information from these scripts. Text lines in handwritten documents are often skewed and close to each other. Because of the low quality and the complexity of these documents (background noise, artifacts due to aging, interfering lines), automatic text line segmentation remains an open research field. It is not easy to extend algorithms for machine printed documents to handwritten documents where text lines are not perfectly straight. Accordingly, handwritten text line segmentation is still a big challenge in document image analysis.

Many methods have been proposed for the text line segmentation over the last decades. But most of the methods are restricted to scanned images of machine printed text. They fail to handle camera captured image and handwritten documents. For example, the Docstrum method [3] was based on the assumptions that there is a global skew angle and the scale of text is almost constant in a given document, which is not true for many challenging cases.

The Text lines provided by segmentation process extracts isolated lines from a document image. Such segmented text lines are generally included in handwriting databases. The segmentation process is relatively easy when text lines are straight, sufficiently spaced, and oriented in the same direction. However, when ascenders and descenders in the inter-line space are present, or when skew is different from one line to another, fragments from neighbouring lines may appear in the background of text-line images. They are considered as noise. Handwritten text lines are also characterized by fluctuations. Due to writer movement, changes in baseline position occur along the text line. The baseline, the fictitious line which follows and joins the lower part of the character bodies, may be straight, straight by segments, or curved [5]. So it is required to consider all the problems for efficient text line segmentation.

This work presents a new methodology which considers the interaction between the text lines and the regional property of each text line. Precisely, the problem is formulated as an energy minimization problem so that the minimization of the cost function yields a set of text lines. Text-line extraction is considered as a grouping problem of CCs and a cost function is developed. Thus text-lines are extracted from a document by minimizing the cost function. Then it is required to detect and correct the skew of the text line. This is done by normalising the lower baseline to a horizontal line using skating window approach. This approach copes with baselines which are skewed, fluctuating, or both. This approach differs from machine learning approaches which need manual pixel assignments to baselines.

II. METHODOLOGY

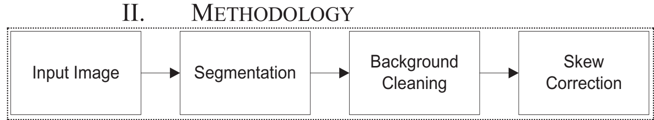


Figure 1. Block Diagram of the Proposed System

Figure 1 gives the overview of the proposed system. A handwritten text document undergoes segmentation whose output is given to the background cleaning. In this stage all the noise is removed. Then the skew is detected and corrected in skew correction stage. The text line thus obtained is free from skew. Each stage is described in detail in the remaining part of the paper.

978-1-4799-2572-8/14/$31.00 ©2014 IEEE

III. SEGMENTATION

In segmentation stage, the input undergoes pre-processing

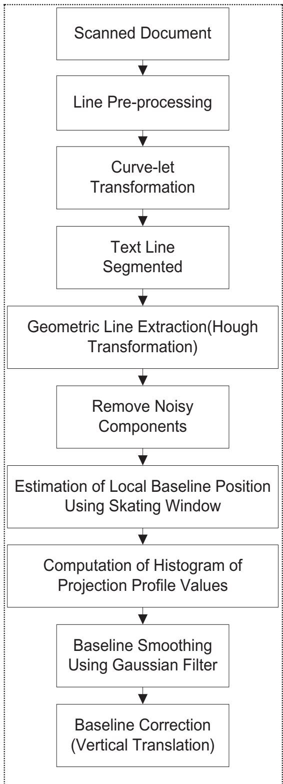


Figure 2: Flow Diagram of the Proposed System

steps and the connected components are found. These connected components undergo bottom-up grouping, for the purpose of energy minimization.

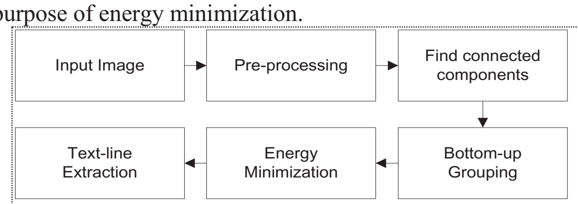


Figure 3: Block Diagram of the Segmentation Stage

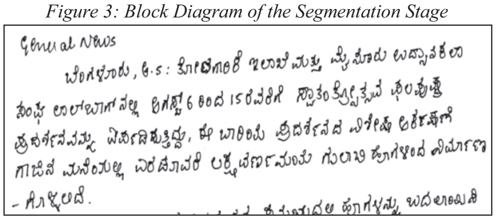


Figure 4: Handwritten Document

A. Energy-Based Formulation of a Text-Line Extraction Problem

Here, energy function is presented whose minimization yields a set of text lines [1]. The function comprises of two terms, and its minimization naturally allows the consideration of the local and global properties of text lines.

1) Energy Function

Text-line extraction can be formulated as a segmentation problem of CCs: $\mathcal{P}$ is divided into several clusters and each cluster is considered as a string line (a string line from the cluster can be constructed). Precisely, text-line extraction is equivalent to a process that finds

$$\mathcal{W} = \{C_1, C_2, \dots, C_K\} \tag{1}$$

where,

$$C_1 \cup C_2 \cup \dots \cup C_K = \mathcal{P} \tag{2}$$

and $C_i \cap C_j = \emptyset (1 \le i < j \le K)$. Based on this observation, text line extraction problem is formulated as

$$\widehat{\mathcal{W}} = \arg\min_{\mathcal{W}} E(\mathcal{W}) \tag{3}$$

where the cost function comprises of two terms, i.e.,

$$E(\mathcal{W}) = E_F(\mathcal{W}) + E_D(\mathcal{W}) \tag{4}$$

The first term $\mathrm{E_F}(\mathcal{W})$ reflects the fitting error of each text line (local property), and the second term $\mathrm{E_D}(\mathcal{W})$ considers the distances between text lines (layout property). In designing the cost function, each segment is considered as a polynomial function representing its center line, and $E(\mathcal{W})$ is calculated in terms of this set of fitting functions.

2) Fitting Functions and Normalized Measures

If line spacing in a given document is static and the spacing is known, it is possible to normalize the fitting error and interline distance. The normalization method presented here is based on this observation: the line spacing is measured and this value is used in normalization. The line spacing value varies according to not only documents but also its location even in the single document as shown in Figure 4. Therefore, line spacing is estimated locally by using the method in [4], rather than assuming that line spacing is a global constant. The $\mathrm{s_p}$ values of some CCs in Figure 4 are illustrated in Figure 5(c). Spatially varying line spacing can be reliably estimated, as shown.

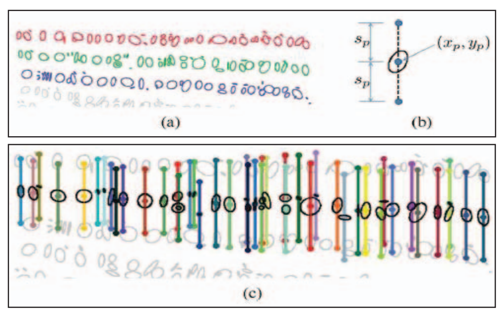


Figure 5: (a) Super-pixel representation (ellipse approximation) of Figure 4 Illustration of (c) Estimated line spacing of CCs in the second line of (a).

a) Normalized Fitting Error:

The fitting function of a cluster C is given by

$$f_C = \arg\min_{f \in P[x]} \sum_{p \in C} (y_p - f(x_p))^2 \tag{5}$$

where (x_p, y_p) is the center of the pth CC, and $P[x]$ is a set of first- and second-order polynomials. Unlike the cases of camera captured images [4], text lines are less curved in handwritten cases, and function f_C is found among first- or second-order polynomials. Then, the fitting error of a cluster C is given by a root-mean-square error

$$\eta(C) = \sqrt{\frac{1}{|C|}} \sum_{p\in C}\left(y_p - f(x_p)\right)^2 \tag{6}$$

where $|C|$ is the number of elements in C. Finally, our normalized fitting error is defined as

$$\eta_n(C) = \frac{\eta(C)}{s(C)} \tag{7}$$

by using the estimated line spacing $s(C) = 1/|C| \sum_{p\in C} s_p$. Since s_p is the estimated line spacing around p, its average value can be regarded as a line spacing around a cluster C. For example, if C corresponds to the second line in Figure 5(a), $s(C)$ is half of the average height of the drawn line segments in Figure 5(c). In [4], they found out that most text lines satisfy

$$\eta_n(C) < 0.2 \tag{8}$$

and C is called curvilinear when it satisfies this inequality.

b) Normalized Distance Between Text Lines:

The distance between two clusters is the minimum distance between their corresponding fitting functions (5), as illustrated in Figure 6. However, the definition becomes a little complicated since these functions may not have a common x axis interval.

First the interval of a cluster C is defined as

$$I(C) = \left[\min_{p\in C} x_p - \delta, \max_{p\in C} x_p + \delta\right] \tag{9}$$

where is the average width of CCs in a cluster C. When two clusters ($C_i$and C_j) have some common intervals, the distance between them is given by

$$\mathrm{dist}(C_i, C_j) = \min_{x\in I(C_i)\cap I(C_j)} \left|f_{C_i}(x) - f_{C_j}(x)\right| \tag{10}$$

Otherwise, the distance is defined as

$$\mathrm{dist}(C_i, C_j) = \left|f_{C_i}(\hat{x}_1) - f_{C_j}(\hat{x}_2)\right| \tag{11}$$

where, $(\hat{x}_1, \hat{x}_2)$ is the closest point pair on the x -axis, i.e.,

$$(\hat{x}_1, \hat{x}_2) = \underset{x_1\in I(C_1), x_2\in I(C_2)}{\arg\min} |x_1 - x_2| \tag{12}$$

The distance with the estimated line spacing is normalised

$$\mathrm{dist}_n(C_i, C_j) = \frac{\mathrm{dist}(C_i, C_j)}{\min\left(s(C_i), s(C_j)\right)} \tag{13}$$

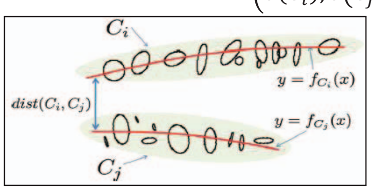


Figure 6: Illustrations of two clustersC_i and, their fitting functions and the distance between them $dist(C_i, C_j)$

1) Design of $E_F(\mathcal{W})$ and $E_D(\mathcal{W})$

Text lines showing small fitting errors are desirable, and the first term $E_F(\mathcal{W})$ in (4) is given by

$$E_F(\mathcal{W}) = \sum_{c\in\mathcal{W}} \beta\left(\eta_n(C)\right) \tag{14}$$

for some increasing function $\beta(\cdot)$. $\beta(\cdot)$ is designed so that the cost difference between two curvilinear clusters ($C_i$and C_j) is small, i.e.,

$$\beta\left(\eta_n(C_i)\right) \simeq \beta\left(\eta_n(C_j)\right) \tag{15}$$

since curvilinear C_i and C_j are already good candidates for text lines. Precisely,

$$\beta(x) \propto e^{-\frac{1}{x}} \tag{16}$$

is set, so that the cost function is almost flat (although it is an increasing function) when $\eta_n(C) < 0.2$.

The second term $E_D(\mathcal{W})$ in (4) reflects the condition that two text lines should not be too close, i.e.,

$$E_D(\mathcal{W}) = \sum_{C_i, C_j \in \mathcal{W}} \gamma\left(\mathrm{dist}_n(C_i, C_j)\right) \tag{17}$$

where

$$\gamma(x) = 1 - \tanh\left(c * (x - x_0)\right) \tag{18}$$

Since $\gamma(x)$ is a soft-thresholding function, this cost function implies that the distance between two text lines is roughly longer than half of the estimated line spacing (i.e., $\mathrm{dist_n}(C_i, C_j) \geq 0.5$). That is, $\gamma(x)$ levies a high cost when the distance between two text lines is much shorter than the estimated line spacing (i.e., $\mathrm{dist_n}(C_i, C_j) \ll 1$), and $\gamma(x) \simeq 0$ when the distance between two text lines is greater than the estimated line spacing. Although the normalized line spacing can be any arbitrary nonnegative values, $\gamma(x)$ is illustrated in a range [0, 1], which is our region of interest (outside the interval, $\gamma(x)$ is almost constant).

c) Curvelet Transform:

The curvelet transform is a multiscale directional transform that allows an almost optimal nonadaptive sparse representation of objects with edges. The currently available implementations of the discrete curvelet transform aim to reduce the redundancy smartly.

In practical implementations, one would like to have Cartesian arrays instead of the polar tiling of the frequency plane. Cartesian coronae are based on concentric squares (instead of circles) and shears (see Figure 7). Therefore, a construction of window functions on trapezoids instead of polar wedges is desirable. For the transition of the basic curvelet according to the new tiling [15], where rotation is replaced by shearing, we use the ansatz

$$\hat{\tilde{\phi}}_{j,0,0}(\xi) := 2^{-3j/4} W(2^{-j}\xi_1) V\left(\frac{2^{\lfloor j/2\rfloor}\xi_2}{\xi_1}\right) \tag{19}$$

with the window function W and with a nonnegative window V with compact support in [-2/3,2/3].

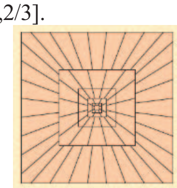


Figure 7: Discrete curvelet tiling with parabolic pseudopolar support in the frequency plane.

B. Minimization of the Cost Function

Since the evaluation of $E(\mathcal{W})$ requires many operations (e.g., curve fitting and distance computation) and the accuracy of $s(C)$ degrades as C deviates from a text line, the minimization of (4) is not an easy job. Hence, a new optimization method is developed [1]: it starts from the coarse solution and refines the solution locally (update two to three text lines at each iteration). This approach not only prevents the large deviation of a cluster C from a text line but also enables the reuse of previously computed values.

II. BACKGROUND CLEANING

The flow of background cleaning is as shown in the Figure 8.

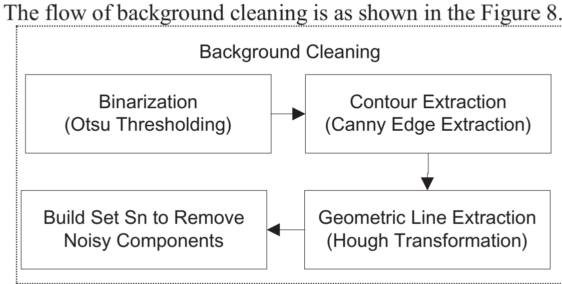


Figure 8: Block Diagram of the Background Cleaning Stage

Text lines result from the segmentation of a document image into rectangular fragments corresponding to text lines. However, extra components from preceding and following lines are often included in text line fragments, as shown in Figure 9(a). The background cleaning [2] comprises of retaining components of the main text line and eliminating noisy components from neighbouring text lines. This process is followed by whitening the background. Nonetheless, cleaning has to be performed with vigilance. As in machine-learning approaches [7], training phase is not needed in this approach. It is based on the extraction of the main text line. Then, components distant from the main text line are eliminated. It may be recognized that a component is qualified as close or distant from the main line, through a threshold directly extracted from writing characteristics of the input image. The background cleaning approach comprises of the following steps:

- The text-line image undergoes binarization (Otsu thresholding) and contours are extracted (by a Canny edge extractor). A geometric line is then extracted from a Hough transform on the contour image. This line corresponds to the highest peak in the Hough (ρ,θ) table [continuous line in Figure 9(a)].

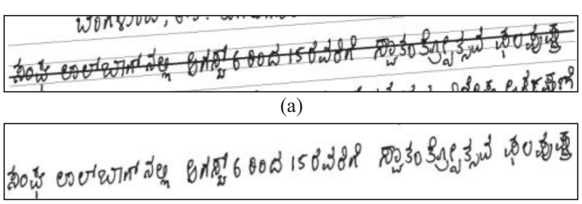


Figure 9: Background cleaning: (a) noisy text line including fragments from two neighboring lines. Gravity centers of connected components (crosses), geometric Hough-based line (continuous) and decision lines (dashed) (b) cleaned line (noisy components removed).

- hmean and hmax, the average and maximum component heights are extracted on connected components which do not reach upper or lower limits of the image.
- An initial set Sc of candidate noisy components is composed of connected components which are distant from the main text line. Their distance is greater than hmean ⁄ 2 computed in Step 2.
- From previous set Sc, set Sn ⊂ Sc is built to remove noisy components. These are Sc components in contact with the upper and lower edges of the image as well as Sc components whose gravity centre is peripheral: the distance of their gravity centre to the main text line is greater than hmax computed in Step 2.

Step 1 ensures locating the main text line regardless of neighboring line fragments or text-line skew. The maximum peak in the Hough (ρ,θ) table always exists and corresponds to the actual text line. Step 2 ensures that the average height of writing components is computed from components which were not cropped by the text-line segmentation process. Step 3 ensures that components that belong to the main text line are not removed even if they do not intersect the geometric Hough-based line. The distance is computed as the nearest distance of the components' pixels to the main text line. Step 4 ensures that peripheral components may be removed even if they do not touch the edge of the image.

Components, classified as noise, are eliminated from the original gray scale image as in Figure 9(b). Residual errors of background cleaning may consist of removing accents and comas from neighboring text lines which are too distant from these lines. Because of scanning and image compression, variations in background intensities appear on images. Even if it is not disruptive for human reading, it can have an impact on features' values. To remove such variations, the background is whitened while preserving grey level foreground values. Thus, features based on grey level intensities can still be extracted. Background pixels are retrieved using the Otsu thresholding method and saturated to value 255.

III. SKEW CORRECTION

Baseline correction is a main step for robust handwriting recognition. It is a necessary step for normalizing handwriting components such as descenders, ascenders, and text body. It also improves feature extraction when features depend on baseline positions. The novel approach, proposed here, for baseline correction, copes with both skew and fluctuation. It is based on a skating-window approach which takes into account both inter-word and intra-word skew.

Baseline correction has been addressed mostly at word level [2] and is tightly linked to de-skew. De-skew consists of estimating a global skew angle on the word image and rotating the word according to this skew angle. Thus, word baseline is corrected through the global rotation of the word. Word de-skew methods are based on linear regressions performed on sets of points. These points may be the centres of mass of word components or the lowest points of the word excluding descenders [8-10]. Word-based de-skewing approaches can be extended to text-lines when lines are straight along one direction as in Figure 10(a), thus, assuming a single skew angle for each line [11]. This assumption is valid for forms or paper sheets including guidelines or for ancient documents when scribes traced marks with lead points. However, this assumption is no longer true for freestyle handwritten documents, since words can be skewed differently alongside text line as in Figure 10(b). Moreover, a handwritten line can be seen quite straight if considered globally, whereas, at word level, baseline position varies greatly.

There is one approach for baseline correction specific to text-line level. It consists of splitting a text line image into vertical strips. A single skew angle is assumed within each strip and a word-based de-skewing approach is applied to each strip based on linear regression [7, 12, 13]. A refined approach consists in selecting the extrema points on which the regression is applied through classification [7]. A trained neural network classifies contour points as belonging to the lower baseline or not. However, this machine-learning approach needs training from labelled samples which is time consuming due to data preparation.

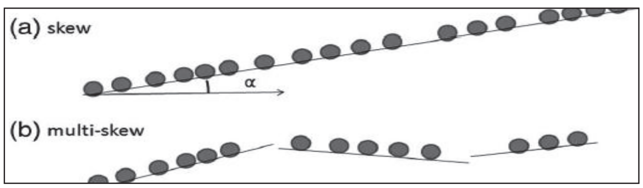


Figure 10: (a) Skewed baseline: bottom positions of writing components lay on a line with angle α ≠ 0. (b) Multiskewed baseline.

In this work, an efficient baseline correction approach is proposed, which is explicit to free-style text-line images. De-skew is performed together with baseline correction. This approach does not require preliminary tasks such as text-line segmentation into words, connected components detection, or run length analysis [14]. The principle of this approach is to estimate the lower baseline position [6] for each image column, by a skating window approach, and correct it by a vertical shift.

This approach starts with a skating window of size w_e which scans the text line image from left to right as shown in Figure 11. The analysis window used, copes with descenders and blank spaces efficiently. For the central pixel of the skating window, the local baseline y position is computed using the projection profile approach [10]:

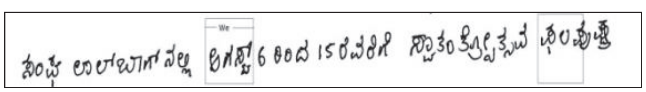


Figure 11: Skating window approach: a window is shifted from left to right along the text line. Within each window, local baseline position is estimated.

- Firstly, the number of foreground pixels are counted along the horizontal direction, at each y position , and the vertical projection profile (PP) is created at that position, as shown in Figure 12(a).
- Second, the distribution (histogram) as in Figure 12(b) of these projection values is computed. This histogram is expected to have a principal mode corresponding to the text-line core zone since the core zone is assumed to have a higher pixel density than the zones corresponding to ascenders and descenders.

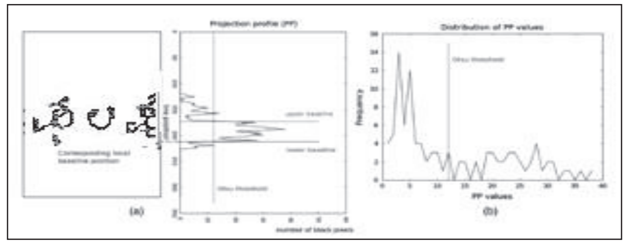
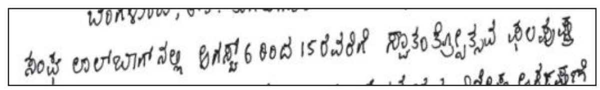


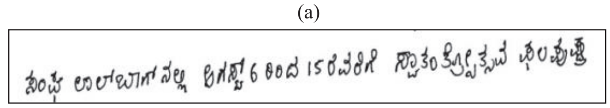
Figure 12: (a) Projection profile (PP) is obtained by a projecting pixels on the vertical axis along horizontal direction. (b) Distribution of PP values and Otsu threshold. Threshold value is then used on PP to extract lower and upper baseline positions.

- Third, a thresholding algorithm (Otsu's method) is applied to the above distribution. The text core zone corresponds to the largest continuous core zone above threshold on the PP profile.

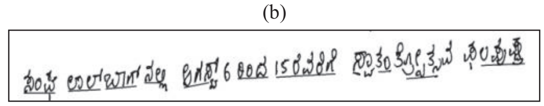
Baseline position curve is then smoothed with a Gaussian filter of width w_{smooth} to get rid of discontinuities. Gaussian distribution standard deviation is $\sigma = w_{smooth}/(4\sqrt{2})$. Without this smoothing, shearing artefacts can appear on corrected images. The correction step corresponds to a vertical translation.



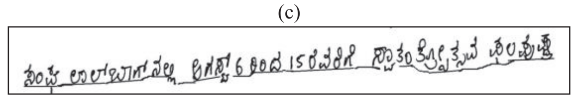
(a)



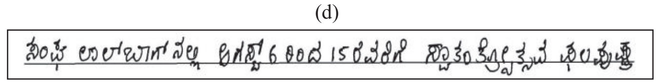
(b)



(c)



(d)



(e)

Figure 13: Pre-processing steps: (a) original segmented line, (b) background cleaning, (c) local baseline estimation, (d) baseline smoothing, (e) baseline correction

IV. RESULT AND DISCUSSION

In order to assess this method, experiment is conducted on a scanned handwritten Kannada documents. The document undergoes binarization as a part of pre-processing, as shown in Figure 14. Then the noise is removed based on the area for the accurate detection of text lines, as in Figure 15. The text lines are detected and segmented; each detected line is indicated by a bounding box, as illustrated in Figure 16. Further, skew of each of the segmented line is detected and corrected, provided that the skew angle of the text line is less than 10°. The skew corrected output is as shown in Figure 17.

CONCLUSION

This work presents a robust scheme of extracting text lines from a scanned handwritten Kannada document image. A cost

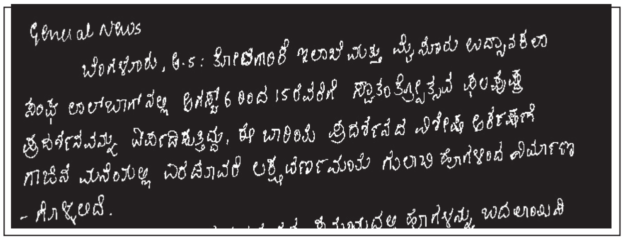


Figure 14: Binarisation of the Text Image

Figure 15: Area Based Noise Removal

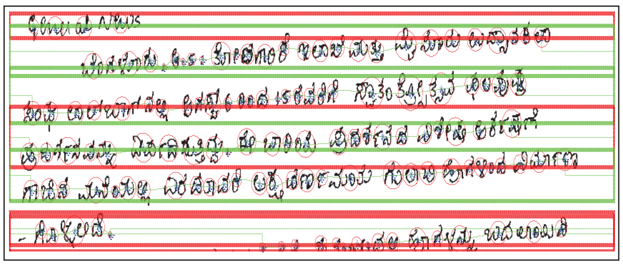


Figure 16: Segmentation Result of Handwritten Kannada Document

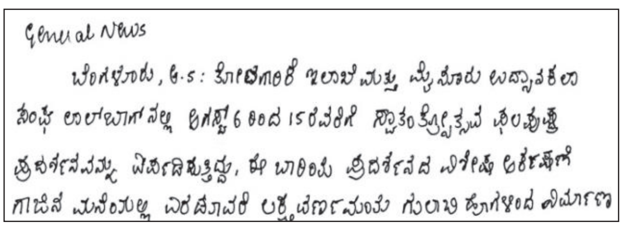


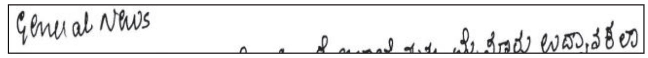
Figure 17: Text Document after Skew Correction Stage

TABLE I. PERFORMANCE EVALUATION OF THE PROPOSED SYSTEM

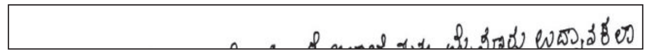
Number of Documents	80
Average Number of Lines in Document	20
Total number of Lines Analysed	1600
Number of Lines Segmented	1586
Number of Skew Corrected Lines	1562
Accuracy of Segmentation	99.13%
Accuracy of Skew Correction	97.63%

function is used, that considers both the fitting error of each text line and the distances between text lines. This work also uses skew correction of the extracted text-line. The skew is corrected using baseline correction method. Here, a skating window is used for the local baseline estimation, thus, avoiding segmentation of text line into subparts. Then the estimated baseline is smoothed and de-skew is performed jointly with baseline correction. This method is proved to be very efficient compared to the traditional methods.

The system eliminates small text fragments in the background cleaning stage, as illustrated in Figure 18. This issue is taken care in the future work.



(a)



(b)

Figure 18: (a)Small Text Fragment along with Components from Neighbouring Text Line, (b)Small Text Fragment Eliminated.

REFERENCES

[1] H. I. Koo and N. I. Cho, "Text-Line extraction in handwritten Chinese documents based on an energy minimization framework," IEEE Transactions on Image Processing, Vol. 21, no. 3, March 2012.

[2] Olivier Morillot, Laurence Likforman-Sulem and Emmanuèle Grosicki, "New baseline correction algorithm for text-line recognition with bidirectional recurrent neural networks," Journal of Electronic Imaging 22(2), 023028 (Apr–Jun 2013).

[3] L. O'Gorman, "The document spectrum for page layout analysis," IEEE Trans. Pattern Anal. Mach. Intell., vol. 15, no. 11, pp. 1162–1173, Nov. 1993.

[4] H. I. Koo and N. I. Cho, "State estimation in a document image and its application in text block identification and text line extraction," in Proc. ECCV (2), 2010, pp. 421–434.

[5] L. Likforman-Sulem, A. Zahour, and B. Taconet, "Text line segmentation of historical documents: A survey," Int. J. Doc. Anal. Recognit., vol. 9, no. 2, pp. 123–138, Apr. 2007.

[6] M. Schambach, J. Rottland, and T. Alary, "How to convert a Latin handwriting recognition system to Arabic," in Int. Conf. Frontiers in Handwriting Recognition, IEEE Computer Society Press, New York (2008).

[7] S. España-Boquera et al., "Improving offline handwritten text recognition with hybrid HMM/ANN models," IEEE Trans. Pattern Anal. Mach. Intell. 33(4), 767–779 (2011).

[8] M. M. Blumenstein, C. K. Cheng, and X. Y. Liu, "New preprocessing techniques for handwritten word recognition," in The 10th IASTED International Conference on visualization, Imaging, and Image Processing, pp. 480–484, Acta Press (2002).

[9] R. Bozinovic and S. Srihari, "Off-line cursive script word recognition," IEEE Trans. Pattern Anal. and Mach. Intell. 11(1), 68–83 (1989).

[10] A. Vinciarelli and J. Luettin, "A new normalization technique for cursive handwritten words," Pattern Recogn. Lett. 22(9), 1043–1050 (2001).

[11] U.-V. Marti and H. Bunke, "Handwritten sentence recognition," in Proc. Int. Conf. on Pattern Recognition, pp. 467–470, IEEE Computer Society Press, New York (2000).

[12] A. Vinciarelli, S. Bengio, and H. Bunke, "Offline recognition of unconstrained handwritten texts using HMMs and statistical language models," IEEE Trans. Pattern Anal. Mach. Intell. 26(6), 709–720 (2004).

[13] M. Liwicki et al., "A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks," in Proc. 9th Int. Conf. Document Analysis and Recognition, pp. 367–371, IEEE Computer Society Press, New York (2007).

[14] Z. Shi and V. Govindaraju, "Skew detection for complex document images using fuzzy runlength," in Int. Conf. on Document Analysis and Recognition, pp. 715–720, IEEE Computer Society Press, New York (2003).

[15] Jianwei Ma and Gerlind Plonka "The Curvelet Transform" IEEE Signal Processing Magazine [118] March 2010.