2011 International Conference on Document Analysis and Recognition

A New Text-line Alignment Approach Based on Piece-wise Painting Algorithm for Handwritten Documents

Alireza Alaei
Department of Studies in Computer Science, University of Mysore, Mysore, 570006, India
alireza20alaei@yahoo.com

P. Nagabhushan
Department of Studies in Computer Science, University of Mysore, Mysore, 570006, India
pnagabhushan@hotmail.com

Umapada Pal
Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata-108, India
umapada@isical.ac.in

***Abstract*—Because of writing styles of different individuals, some of the text-lines may be curved in shape. For recognition of such text-lines, their proper alignment is necessary. In this paper, we propose a text-line alignment technique based on painting algorithm. Here at first, Piece-wise Painting Algorithm (PPA) is used to get a number of black and white rectangular patches all along the text-line for text-line alignment. Identifying the degree of oscillation of the input text-line, some candidate pixels are also obtained based on horizontal projection and center points of the black patches. Using the degree of oscillation of the input text image and the candidate pixels a curve or straight line is fit to trace the baseline. Subsequently, all components of the text-line are deskewed based on analyzing the characteristic of the fit curve or line to align the components with respect to the horizontal imaginary baseline. The proposed algorithm was evaluated with 128 Persian handwritten text-lines containing 4317 subwords. Experimental analysis showed that 92.31% of the subwords were accurately aligned. Further, the proposed algorithm was tested with another Persian handwritten text-lines dataset [6] and remarkable results were achieved.**

Keywords-Handwritten Recognition; Textline alignment; Piece-wise Painting Algorithm; Curve Fitting; Linear Regression

I. INTRODUCTION

In any OCR system, preprocessing is the primary footstep that plays an important role to obtain higher recognition accuracy. Skew detection is one of the preprocessing steps used to correct skew angle of a document [1]. The skew detection techniques [2, 3] cannot be applied when (i) the text-lines of a document are multi-oriented (Figure 1(a,b)), (ii) the text-lines are wavy as shown in Figure 1(c) or (iii) the text-lines are curved in shape (Figure 1(d)). Proper alignment of such text-lines is necessary before recognition.

Baseline information is useful in different stages of Persian handwritten recognition [4, 5]. Recently two algorithms for alignment of Persian/Arabic handwritten text-line [6] as well as for estimating the baseline for Arabic handwritten word/subword were presented [7]. The first stage of the work presented in [6] estimates the writing path of a text-line by a curve fitting method using candidate baseline pixel points detected using a template-matching algorithm. In the second stage, slant and position of components in the text-line is obtained using fit curve information and center points of the components. Next, the baseline for each subword is corrected. In [7] an algorithm for Arabic baseline estimation in subword level is proposed. First small diacritics are eliminated from the text image. Then using a thinning algorithm skeleton of the image is obtained. Some cross and branch points of skeleton image are extracted from horizontal mid-band of the image. Finally, by using a linear regression on the local minima of lower contour and linear interpolation on support points of extracted regression lines, the baseline is extracted. The technique presented in [7] cannot be used for estimating baseline in multi-oriented, curved or wavy Persian/Arabic text-line. The template matching technique presented in [6] results in improper candidate pixels in many cases, which consequently provides inappropriate baseline alignment. Moreover, both the techniques [6, 7] are time-consuming because of using template matching and thinning operation.

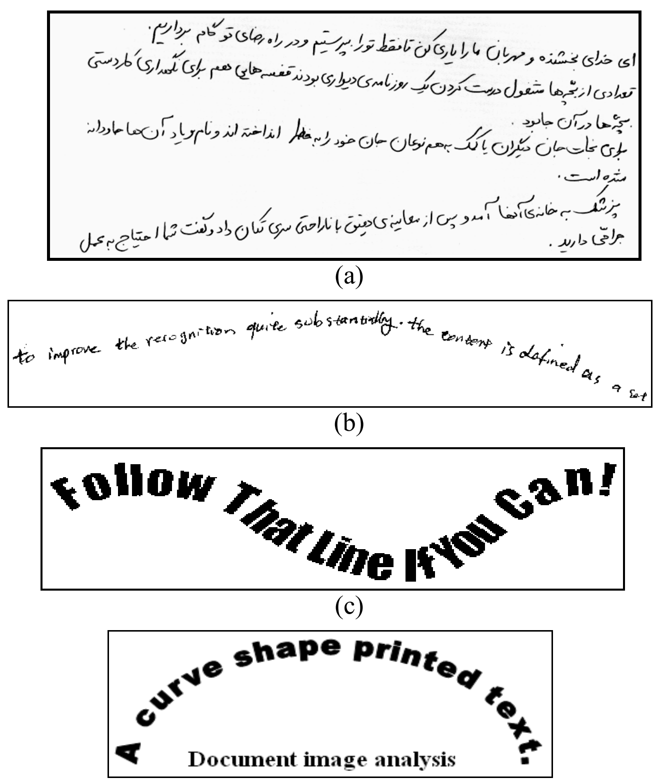

Figure 1. (a) A multi-skew Persian handwritten document and 3 other types of handwritten/printed samples of English (b-d).

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.73

In this paper, we propose a Piece-wise Painting [8] based framework for handwritten text-line alignment of Persian, Arabic, and Urdu scripts. The proposed approach can also be used in other scripts such as English. The Piece-wise Painting Algorithm (PPA) can take care of the skew as well as oscillatory property of a text-line [8]. In the proposed technique, the PPA is employed to prepare rectangular patches of black blocks all along the text-line. Subsequently, center points of the black patches are selected and based on these points degree of oscillation of the input text-line is determined. Based on the degree of oscillation, it can be notified that a text-line is a straight/skewed text-line or a curve/wavy shape text-line. In a straight/skewed text-line, best straight line is selected and its slope is defined as estimated skew of the text-line. In an oscillatory text-line, a slope for each component is calculated based on analyzing the characteristic of fit curve. Each component presented in the text-line is then rotated by respective slope to approximately align it with the horizontal line. Further, each component is lined up to achieve more accurate alignment (horizontal line) for the text-line.

II. PROPOSED TEXT-LINE ALIGNMENT TECHNIQUE

Different stages in the block diagram of the proposed text-line alignment technique are shown in Figure 2. Details of each stage are illustrated in the following subsections.

A. Piece-wise Painting Algorithm (PPA)

The Piece-wise Painting Algorithm (PPA) is a novel idea introduced in [8] and used for handwritten text-line segmentation. In this research work, we extended the concept of painting for baseline detection and text-line alignment in a scanned document.

To trace the baseline it is necessary to find points very close to the baseline (reference line). To do so, the input text-line (gray or binary image) is converted into a painted image by employing the algorithm presented in [8]. This algorithm [8] works based on decomposing the image into a number of vertical stripes with a certain width (average width of components) from the right side (Figure 3). Then every pixel value in each row of the stripe is replaced by a gray intensity, which is the average intensity value of all pixels present in that row of the stripe (Figure 4). Subsequently, the gray stripes are converted into binary. Details of the PPA can be found in [8].The result of this step, which is an image with a number of rectangular black blocks, is shown in Figure 5.

B. Finding the degree of oscillation

To have a single black rectangular area in each stripe of the painted image representing text portion of a line, the painted image should be smoothed. Therefore, for every column the upper and lower most black pixels are found and the white pixels, if any, between the upper and lower most black pixels are converted into black. As a result, in each stripe a single smoothed black rectangular area is obtained (see Figure 6). The black rectangular areas in the smoothed painted image help to find points (pixels) close to the baseline as well as degree of oscillation of the text-line.

In [6], degree of oscillation of any text-line was regressed to 3 for baseline estimation This predefined value will not work for baseline estimation in the cases that a straight line or a curve with degree of less than 3 can be fit to the input text image. Hence, in this paper we introduce a new method to find the degree of oscillation of every text-line automatically. To do so, the center points of all black rectangular areas are chosen. A polynomial curve is drawn based on the list of these center points (Figure 7). The curve is then smoothed and all local maxima and minima are identified (Figure 8). The sum of number of true local maxima and minima is defined as degree of oscillation of the input text-line. If the distance between a maximum and a minimum is greater than WI/6, we call that maximum/minimum a true maximum/minimum. WI is width of the input text-line and the value of WI/6 is decided by experiment. Since, this heuristic value is obtained based on the smoothed image (Figure 6), it is independent to handwritten scripts. Moreover, it can support maximum 6 oscillations in a input text-line, where in a real situation, we have not observed more than 5 oscillations in a text-line. From Figure 8 it is evident that there are 2 true maxima and 1 true minimum. Therefore, degree of oscillatory of the input text image is 3. It can also be noted that if there is no local minimum and maximum, the degree of oscillation of the input text-line will be considered 0. In other word, entire text-line suffers only from a single skew angle.

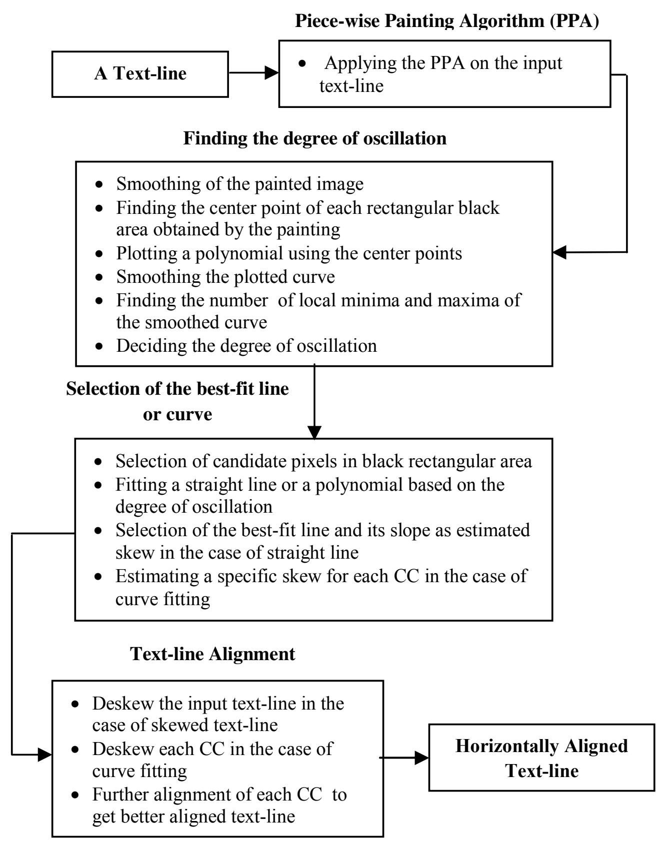

Figure 2. Block diagram of the proposed text-line alignment technique

C. Selection of the best-fit line or curve

Based on the degree of oscillation of a text-line, a line or a curve fitting is performed. If the degree is one or more, then a curve is fit to the text-line and it is considered as its baseline. Otherwise, a straight line is fit to the input text-line and it is considered as its baseline.

To fit a curve on the input text-line, first the text-line is mapped on the smoothed painted image. For each black rectangular area, the corresponding image part in the original image is found and the piece-wise horizontal projection is computed for the original image portion. The row, where the peak of the histogram lies, is noted and the black pixels on that row of the original image portion are the candidate pixels for that black rectangle area. By knowing the degree of oscillation and also having the candidate pixels, a polynomial fitting algorithm [9] is utilized to fit a curve to the input text-line. Formula for polynomial with degree of n is shown in the following.

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_2 x^2 + a_1 x + a_0 \quad (1)$$

Where n is the degree of polynomial (nonnegative integer) and a_0, a_1, a_2, ..., a_n are constant coefficients and $f(x)$ is defined for all values of x. In the present work the (x, y) coordinates of candidate pixels are mapped with $(x, f(x))$ in equation 1, respectively. The degree of polynomial (n) is considered as the degree of oscillation and the polynomial coefficients are calculated accordingly. The result of curve fitting operation is shown in Figure 9. This curve is further used to align the oscillatory written input text-line into a straight horizontal text-line, which facilitates subsequent processes. Figure 10(a) shows the fit curve to a Persian printed wavy text-line.

To find the best-fit straight line for the input text-line, a simple and efficient technique is proposed. The center points on top, middle and bottom of each black rectangular area are considered for the purpose. As a result, three groups of points are chosen from tops, middles, and bottoms of black rectangular areas, respectively. For each group of points, two different straight lines are drawn. The first line is drawn using linear regression concept and the other line is drawn connecting the left most point to the right most point of that group. As a result, six different lines are fit to the input text-line. Using the candidate pixels and employing linear regression another straight line is also fit. Amongst these seven lines, one is considered as the best-fit line using the following criteria.

$S_r = Round\,(Absolute(S))$ *(2)*

$S_M = Mode\,(S_r)$

$S_d = Absolute(S - S_M) = Absolute(s_1 - S_M, s_2 - S_M, s_3 - S_M, s_4 - S_M, s_5 - S_M, s_6 - S_M, s_7 - S_M)$

Estimated skew angle $= S\,(Minimum\,(S_d))$

where $S = (\mathbf{s_1,s_2,s_3,s_4,s_5,s_6,s_7})$ is the list of 7 slopes (skew angles) obtained for a given text-line, $S_r = (\mathbf{s_{r1},s_{r2},s_{r3},s_{r4},s_{r5},s_{r6},s_{r7}})$ is rounded values of absolute of the list S, S_M is statistical mode of the values in S_r and $S_d = (\mathbf{s_{d1},s_{d2},s_{d3},s_{d4},s_{d5},s_{d6},s_{d7}})$ is absolute of differences between the values of S and S_M. The angle having minimum difference with S_M represents the skew angle of the given text-line and consecutively the corresponding straight line is the best-fit straight line (baseline). Figure 10(b) shows the best-fit straight line on a Persian printed text-line with a upward skew. More results of baseline tracing based on curve as well as straight-line fitting are shown in Figure 10(c-e).

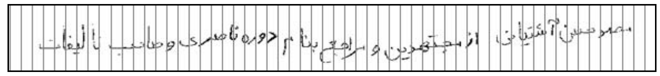

Figure 3. A Persian text-line image after decomposing it into vertical stripes.

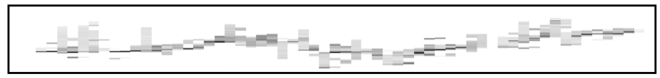

Figure 4. Gray level painted image after applying painting algorithm on image shown in Figure 3.

Figure 5. Binary image obtained after applying binarization on the image shown in Figure 4.

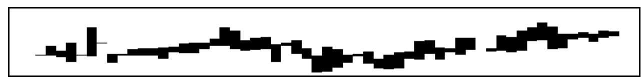

Figure 6. Smoothed image obtained from Figure 5.

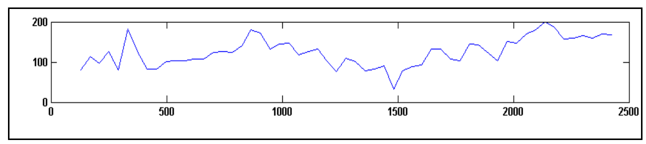

Figure 7. Polynomial curve obtained from center points of the rectangular black areas of the image shown in Figure 6.

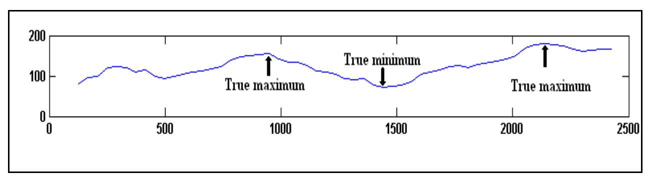

Figure 8. Smooth Polynomial curve obtained from center points of the rectangular black areas of the image shown in Figure 7

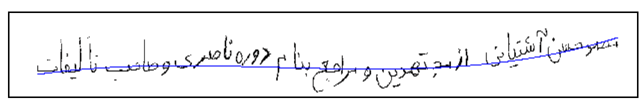

Figure 9. Baseline (curve fitting result marked in blue color) obtained for the text-line shown in Figure 3.

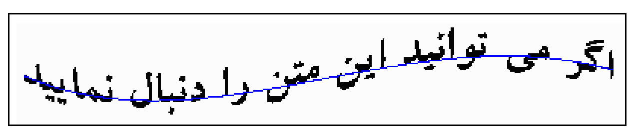

Figure 10(a). Baseline (marked in blue color) obtained for a Persian printed wavy text-line.

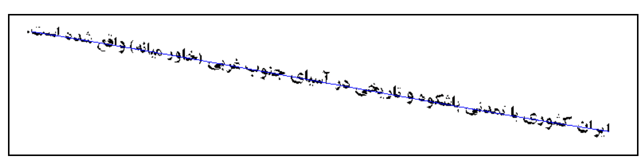

Figure 10(b). Baseline (marked in blue color) obtained for a skewed (upwards) Persian printed text-line.

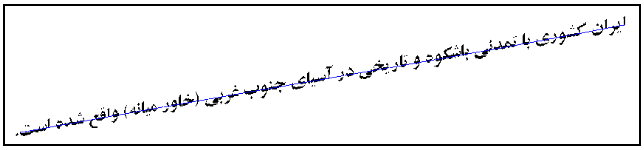

Figure 10(c). Baseline (marked in blue color) obtained for a skewed (downwards) Persian printed text-line.

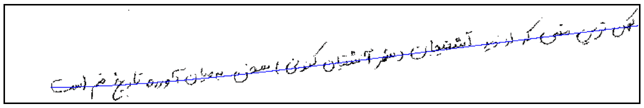

Figure 10(d). Baseline (marked in blue color) obtained for a Persian handwritten text-line.

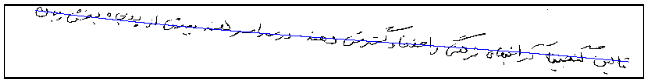

Figure 10(e). Baseline (marked in blue color) obtained for a Persian handwritten text-line.

D. Text-line alignment

Based on the number of oscillation in an input text-line a curve or a straight line is fit for baseline tracing. Therefore, for text-line alignment also two different strategies are proposed. In the case of curve line, different pixels in a component should actually be rotated with different slopes. Therefore, for every connected component, a single slope is calculated and every component is rotated with its own slope to align with reference to the horizontal axis. To find the slope for each component in the text-line all intersection points, which occur between the fit curve and the components (words, subwords, dots and strokes) presented in the input text-line, are obtained. Since a curve can be considered as a collection of partially linear pieces, the right most and the left most intersection points of the fit curve with a particular component are connected to make a piece of line. Figure 12 shows zoomed version of the connected component marked by circle in Figure 11 and its intersection with the fit curve. The slope of this piece of line with respect to the horizontal axis is considered as the orientation of that component. The slope of this line (βi) is used for rotating the component in reverse direction (Figure 13). In addition, there may be a number of components (a component is marked by a rectangle in Figure 11) that do not have any intersection with the fit curve in the text-line. Such components are rotated based on the corresponding slope of their nearest component, which was rotated previously. Figure 14 shows the result of the first-stage of the proposed baseline alignment.

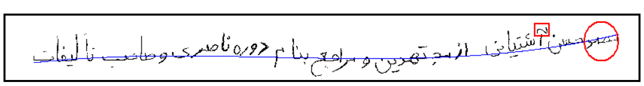

Figure 11. One of the components having intersection with the fit curve is indicated by a circle and a component that does not have any intersection with the fit curve is marked by a rectangle.

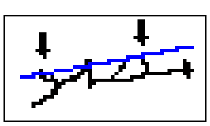

Figure 12. A component and its left most and right most intesection points with the fit curve.

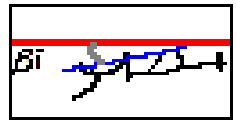

Figure 13. Slopes βi between the horizontal line and line drawn by connecting the two intersection points shown in Figure 9.

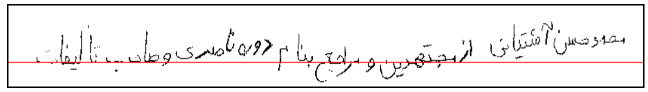

Figure 14. Result of baseline aligning technique after aligning the component in first step.

As it is demonstrated in Figure 15, some components are not placed in proper positions with respect to the horizontal baseline. Therefore, it is necessary to shift those components to the appropriate position to get more accurate baseline. To do so, for every component of the input text-line, horizontal projection is computed and maximum horizontal projection value and its position are found. Since the baseline of a component may place in the position of maximum horizontal projection or end-points (uppermost/lowermost), each component having at least one intersection point with the fit curve (baseline) is vertically shifted upward or downward to align with reference to horizontal baseline. The difference between the position of the horizontal baseline and the position of maximum horizontal projection or end-points (uppermost/lowermost) of each component is measured as value for shifting a particular component. If the shifting value is less than the average component height then the movement will take place. The components having no intersection point with the drawn baseline are vertically moved based on the corresponding movement (up/down) of its nearest component that has been shifted (up/down) previously.

Result of the proposed algorithm on the image shown in Figure 3 is shown in Figure 16.

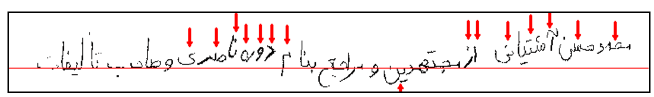

Figure 15. Components which need more alignment with respect to the horizontal baseline are indicated by arrows

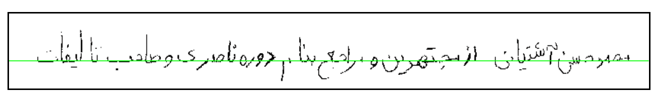

Figure 16. Result of applying the proposed algorithm on image shown in Figure 3

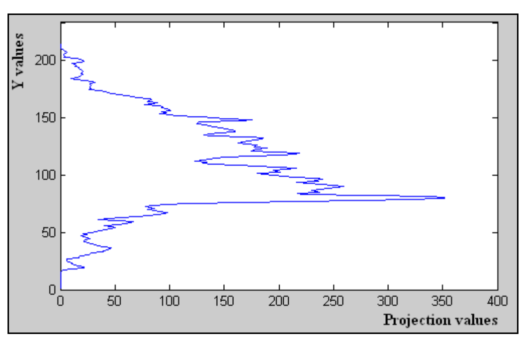

Figure 17. Horizontal projection of the original image shown in Figure 3

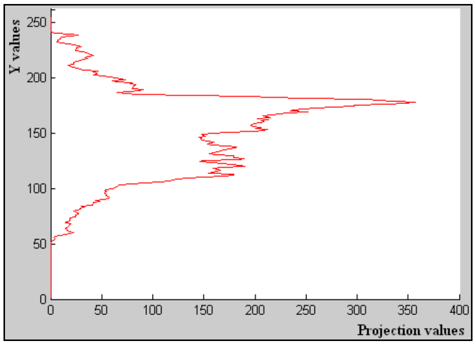

Figure 18. Horizontal projection of the result after aligning the component in first step shown in Figure 14

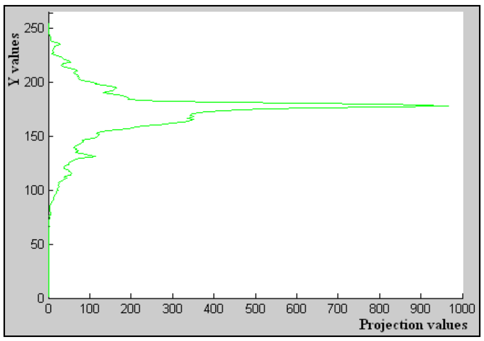

Figure 19. Horizontal projection of the final result shown in Figure 16

III. EXPERIMENTAL RESULTS AND COMPARISON ANALYSIS

We tested the proposed approach using two different datasets. The proposed approach for text-line alignment was first tested on 128 handwritten Persian text-lines containing 4317 subwords written by different writers and scanned in gray level with resolution of 300 DPI. Since, ground truth information for baselines have not been provided, we evaluated the accuracy of text-line alignment approach for each text-line manually. We assumed a word/subword was aligned correctly with respect to the horizontal baseline (peak of horizontal projection), if the vertical distance between the horizontal baseline and the actual baseline of that word/subword was less than 15 pixels. This margin threshold was chosen based on the work presented in [6] to have a fair comparative study. Accuracy of the proposed algorithm is calculated based on dividing the number of correctly aligned words/subwords by the total number of words/subwords in all the text-line images. From experimental results, we noted that 92.31% of the words/subwords were correctly aligned with respect to the horizontal baseline. Figures 17-19 show the results of horizontal projection in original image, image after the first stage of text alignment, and image after the second stage of text alignment, respectively. By looking at these figures, considerable improvements in aligning the text-line can be observed.

Furthermore, we tested the proposed scheme with the dataset introduced in [6]. The dataset consists of 600 text-lines containing 14600 subwords without ground truth information [6]. In subword level, we manually evaluated the present work and 14058 (96.31%) words/subwords out of 14600 subwords words/subwords were correctly aligned with respect to their baselines. The result of the baseline correction method proposed in [6] was 95.12% with the same dataset and the same margin parameter. This shows improvement of our method in the Persian baseline alignment.

Since, text-lines of the first dataset developed by us are more complex than the text-lines of the dataset [6], we achieved lower accuracy (92.31%) employing the proposed approach on the first dataset when compared to the result (96.31%) obtained testing the proposed technique on the dataset used in [6].

To show applicability of the proposed framework on other languages a few lines of printed English text with different contents and different skews (10 to 20 degree) are also considered. A result of the proposed technique on an English text-line is shown in Figure 20.

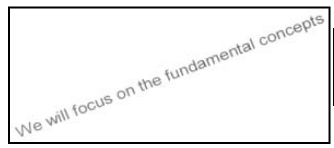

Figure 20 (a). A sample of printed English text-line

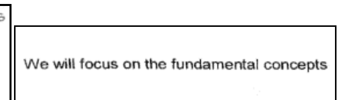

Figure 20(b). Result of the proposed technique obtained from 20-a

IV. CONCLUSION

In this paper, a novel algorithm for text alignment is introduced. The effectiveness of the proposed algorithm is evaluated with different datasets. The proposed algorithm is shown better result compared to the results of recent algorithms. The main advantage of the proposed algorithm is automatically detection of a degree of oscillation for every text-line for proper alignment.

REFERENCES

[1] J.J. Hull, Document image skew detection: survey and annotated bibliography, in: J.J. Hull, S.L. Taylor (Eds.), Document Analysis Systems, Vol. 2, World Scientific, Singapore, 1998, pp. 40–64.

[2] B. Gatos, N. Papamarkos, C. Chamzas, Skew detection and text line position determination in digitized documents, Pattern Recognition 30 (9), 1997, pp. 1505–1519.

[3] C.-H. Chou, S.-Y. Chu, F. Chang, Estimation of skew angles for scanned documents based on piecewise covering by parallelograms, Pattern Recognition 40, 2007, pp.443–455.

[4] A. Alaei, P. Nagabhushan and U. Pal, "A Baseline Dependent Approach for Persian Handwritten", Proceedings of 20th ICPR, pp. 1977–1980, 2010.

[5] R. El-Hajj, L. Likforman-Sulem and C. Mokbel, Arabic Handwriting Recognition Using Baseline Dependant Features and Hidden Markov Modeling, Proceedings of Eighth ICDAR, pp. 893–897, 2005.

[6] M. Ziaratban, K. Faez, A Novel Two-Stage Algorithm for Baseline Estimation and Correction in Farsi and Arabic Handwritten Text line, Proceedings of the 19th ICPR, pp. 1–5, 2008.

[7] H. Boukerma, N. Farah, A Novel Arabic Baseline Estimation Algorithm Based on Sub-Words Treatment, Proceedings of the 12th ICFHR, pp. 335–338, 2010.

[8] A. Alaei, U. Pal, P. Nagabhushan, A new scheme for unconstrained handwritten text-line segmentation, Pattern Recognition 44 (4), 2011, pp. 917–928.

[9] I.D. Coope, Circle fitting by linear and nonlinear least squares, Journal of Optimization Theory and Applications 76 (2), 1993, pp. 381–388.