International Journal of Machine Intelligence
ISSN: 0975-2927 & E-ISSN: 0975-9166, Volume 6, Issue 1, 2015, pp.-426-434.
Available online at http://www.bioinfopublication.org/jouarchive.php?opt=&jouid=BPJ0000231

SEGMENTATION AND ALIGNMENT OF MULTI-ORIENTED AND CURVED TEXT LINES FROM DOCUMENT IMAGES

BOAZ T.K.* AND PRABHAKAR C.J.
Department of Computer Science, Kuvempu University, Shivamogga - 577 451, Karnataka, India.
*Corresponding Author: Email- kipyego3@yahoo.com.au

Received: December 24, 2014; Revised: May 12, 2015; Accepted: May 14, 2015

Abstract- In this paper, we present a novel approach to segment and align multi-oriented and curved text-lines from document images. We assumed that the input document image contains text-lines with arbitrary orientation and identified the arbitrary text string based on projection profile. We employed anisotropic Gaussian filter bank on the identified arbitrary text region in order to smooth the text region, which helps to detect the ridges which is a representative of a text-line path. The ridges are then labeled and a cubic B-spline is fitted to the text-line path points. The orientation and curvature features of the text-line path is estimated using orientated gradients for each point and corresponding curvature to these text-line path are computed. Text is aligned along the horizontally transformed line by rotating individual characters based on the computed curvature information. Finally, the aligned text-lines are extracted, which can be fed into OCR for recognition. The evaluation metrics was evaluated at text-line segmentation level and the results posted show a significant improvement. The resulting system is proven to provide better results than most state of the art algorithms.

Keywords- Text-Lines detection, Text-line segmentation, oriented gradients

Citation: Boaz T.K. and Prabhakar C.J. (2015) Segmentation and Alignment of Multi-Oriented and Curved Text Lines from Document Images. International Journal of Machine Intelligence, ISSN: 0975-2927 & E-ISSN: 0975-9166, Volume 6, Issue 1, pp.-426-434.

Introduction

TEXT-LINE segmentation is one of the most important layout analysis steps in document image understanding system. Huge amounts of historical document images are being published by on-line digital libraries worldwide. However, for these raw digital images to be really useful there is a need to transcribe them into a textual electronic format that would allow unrestricted indexing, browsing and querying. The simulation of human reading has been the subject of intensive research and a significant amount of improvements has been achieved, though humans still outperform even the most powerful computers in relatively routine functions such as vision [1]. Therefore, increasing need for newly emerging development of on-line digital libraries, multimedia databases and systems that require historical handwritten and printed data entry. Document contents have been classified based upon two major criteria, a) the data acquisition process which includes on-line and off-line, and b) the text type comprising of either machine-printed or handwritten text [1]. Handwritten document images forms a large bulk of historical documents and based on available literature on new methodologies and the recent direction of Character Recognition research they are characterized by a number of issues such as, multi-oriented text-lines, high degrees of curl or curved, touching and overlapping components, noisy components, and complex historical writing style, scripts, digitization methods, and intensity values [2,3]. These types of issues with the addition of the need for automatic processing of bulk amount of paper documents and data transfer into machines has drawn intensive research for the development of robust systems capable to provide the said issues with favorable solutions. In the process of acquiring document images, many researchers have used conventional flatbed scanners, where the flatbed type of scanners have a little or no geometric degradation from its original source, ultimately these scanners are proving very inefficient and difficult to use in many areas such as lecture halls and classes as they are bulky and not portable as well not suitable for the automation of historical documents with tightly bound and thick book spine. Images of non-planar paper surfaces such as tightly bound books, page curl causes additional distortion, which poses even greater challenges due to geometry distortions and nonlinear illumination in document image acquisition [4]. Handwritten text-lines in images of thick book's pages acquired with an image scanner are difficult to segment because they are deformed under the influence of curled surface. The introduction of digital cameras and smart phones are proving to be much more user friendly, efficient and flexible, making them possible for acquiring high quality image model of a book surface or a written material, and can be further processed to produce flat output from camera captured documents. Document images captured by digital cameras however have not caught up to flatbed scanners yet, mainly because camera images tend to suffer from geometric and perspective distortions, and are therefore limited in their use for electronic archival or Optical Character Recognition (OCR), unless specialized tools are available.

A large number of text-line segmentation methods are available in the literature [3,5,6], where most of these methods are designed with certain assumptions and fails when these assumptions are not satisfied. Yan et al. [5] applied fuzzy curve tracing algorithm to address the problem of extracting curved text paths in artistic documents. The fuzzy curve tracing initially groups character pixels using the fuzzy c-means (FCM) algorithm, where each cluster center represents all its associated pixels components in a class. A spatial relation among the cluster centers is then analyzed and a final text pixel clustering is done with the constraint that the curve that passes through these cluster centers must be smooth.

Optical Character Recognition (OCR) requires linear text for reading [6]. Therefore optical reading necessitates segmenting the text image into physical base components such as lines, words and characters before performing symbol recognition. To perform these, non -linear text strings must be transformed into linear text in order to introduce readability to OCRs. In the segmentation of text line paths, touching, broken, or overlapping text lines frequently occur and poses a bigger challenge which need to be worked out in order to segment the elements correctly. Handwritten documents have the additional challenge of curvilinear text-line paths [7] which does not guarantee the smoothness of the curve due to availability of ascenders and descenders. An added level of complexity exists since documents have a degree of noise introduced during acquisition time or even from physical damage. Variations such as writing styles in handwritten documents bring new difficulties for segmentation algorithms. Existing methods for text-line segmentations includes projection profiles [6], and the level set methods [8].

In this paper, we present a script-independent text-line segmentation and alignment for handwritten camera-captured document images containing arbitrary multi-oriented, curled or curved text-lines based on estimated orientated gradient and curvature features. A curled text-line is characterized by a progressive variation of orientation angle from positive to negative or vice versa. Text-line segmentation and its subsequent skew correction and sequential alignment along horizontal line are based on estimated curvature of the text-line path. We have used a method that is almost similar to the one presented in [9,10] though our method performs a line-line sequential labeling of text string with the extraction of orientation and curvature feature of the identified text-line path individually. The text-line segmentation is performed by first implementing both line and distance transform based on the extracted features by orientated gradients algorithm.

The rest of the paper proceeds as follows in a sequential order, a review of related work on multi-oriented, curled or curved text-line extraction follows immediately. A detailed discussion of the proposed method comes after the related work, with the experimental results being described just before the conclusion of the paper.

Related Work

Automatic text line segmentation in document images remains an open research field with many techniques developed for both printed and handwritten characters [1,11-14]. In the segmentation of handwritten documents many researchers have presented various approaches that can be classified into two main categories [2]: (a) connected component based and (b) deformable model-based methods in order to address issues relating to the segmentation of handwritten document tasks such as signatures and address blocks on envelopes and mails, Cohen et al. [12] addressed the problem involved in the recognition and comprehension of text image, the authors presented a method that uses bottom-up information to develop a global description that suggests hypothesis to extract semantic information in unconstrained handwritten text in certain limited domain such as postal address, in this type of work the postal addresses may contain only horizontal text-lines. However curled printed or handwritten character in images pose a wide range of challenges and makes it difficult to recognize because they are deformed under the influence of curled surfaces. Hardly any method available in the literature that is dedicated purely to curled text-line extraction in handwritten documents [15,16], though many methods are available that extracts straight multi-oriented printed or handwritten documents. Kasar et al [9] presented a novel method for aligning printed curved character string which can handle multiple text-lines in various layout styles. Yan et al [5] addressed the problem of extracting curved text paths in artistic documents where they applied fuzzy curve tracing algorithm, their method's main drawback is that it cannot handle complex intersecting curves in document images. The fuzzy curve tracing initially groups character pixels using the fuzzy c-means (FCM) algorithm, where each cluster center represents all its associated pixels in a class. A spatial relation among the cluster centers is then analyzed and a final text pixel clustering is done with the constraint that the curve that passes through these cluster centers must be smooth. However, their method's main drawback is that it cannot handle complex intersecting curves in document images. In [17,18], the authors presented methods that employ image meshing for multi-oriented handwritten text-lines. Ouwayed et al, [17] method, was driven by the fact that multi-oriented text-line in historical handwritten documents was caused by writers updating the contents within margins and thus gave rise to lines in different directions and orientations, the image meshing progressively and locally determines the lines and once the meshing has been established, the orientation is then determined by the use of Wigner-Ville distribution on the projection histogram profile. However, their method tend to be applicable mostly only to domains similar to it and not well suited for use in large scale digitization pipelines.

Cruz et al [19] presented a handwritten line segmentation method devised to work on documents composed of several paragraphs with multiple line orientations. The method is based on a variation of the EM algorithm for the estimation of a set of regression lines between the connected components that compose the image. Panwar et al [10] presented a method that performs text-line extraction that uses a connectivity strength parameter with depth first search approach to extract of connected components. Their major drawback is the un-ordered labeling of connected components of the same line in a given document that will require yet another algorithm for classification into different text-lines. Yin et al [20,21] presented a text-line segmentation method that can handle both multi-skewed and curved text-lines based on minimum spanning tree (MST) clustering technique with distance metric learning to group connected components (CCs) into a tree structure based on their distance metric from which text-lines are extracted by dynamically cutting the edges using a hyper-volume reduction criterion and a straightness measure to find the number of clusters. Their first approach was based on hand-crafted distance metric which had some limitations while in their second approach it is based on supervised metric, a methodology that can automatically learn to classify the target components. Pal et al [22] presented a method based on the concept of water reservoir analogy, to address the extraction of individual text lines from printed Indian documents containing multi-oriented and curved text lines. Chiang et al [23] in their paper exploited dynamic character grouping based on character size and maximum desired string curvature for recognition of non-homogeneous text using a variety of raster maps containing multi-oriented, curved and straight text-lines. Yan et al [5] addressed the problem of extracting curved text paths in artistic documents where a fuzzy curve tracing algorithm is applied; the method's main drawback is that it cannot handle complex intersecting curves in document images. The fuzzy curve tracing initially groups character pixels using the fuzzy c-means (FCM) algorithm, where each cluster centre represents all its associated pixels in a class. Vasudev et al [24] introduce an algorithm that transforms arc-form text into linear-form text, in their experimental results the OCR recognizes an arc-form-text as picture rather than text. While in the second experiment they transform arc-form text to linear-form text, these introduce some readability to the OCR system. This obviously makes OCRs to demand linear text for reading.

Our Approach

We have developed a handwritten document processing algorithm which performs the following sequential tasks 1) Text region detection, 2) Text-line path detection, 3) Text-line segmentation and 4) Text-line alignment. The algorithm uses well known robust processing techniques, which are applied sequentially. We employed a projection profiles technique to detect text regions, while rotating anisotropic Gaussian filter bank smoothing with ridge detection techniques are used for text-line detection. We segmented the detected text-lines into individual text-lines by allocating each with individual colors for better visualization and finally text-line alignment has been implemented based on computed orientation and curvature information of each segmented text-line. The above approaches are discussed in details in the following subsections.

Text Region Detection

Given a handwritten document image that we assume to contain image graphics and arbitrary oriented text-lines, then our main interest is to detect regions containing text. To perform this procedure we extracted edges on the given document image as it is assumed that text regions are rich in edge information. We then applied projection profiles technique to robustly identify the text regions. Projection profiles have been adapted by Likforman-Sulem et al., [6] for segmentation of handwritten documents containing slight amount of overlapping text lines. In our work, we adapted projection profiles technique to identify and detect regions rich with text information. The projection graph is characterized by a large and almost constant base area, which is analyzed through observation in order to set the threshold value. The graph should have a minimum compact block base ([Fig-1] red line is the threshold) whose height is much far beyond the threshold value set for arbitrary text-line region detection. The profile curve is then smoothed, using a median filter to eliminate false local alarms and reduce sensitivity to noise [25]. The identification of text region provides a workable environment to apply the favorable algorithms in order to speed the process and reduce computational time.

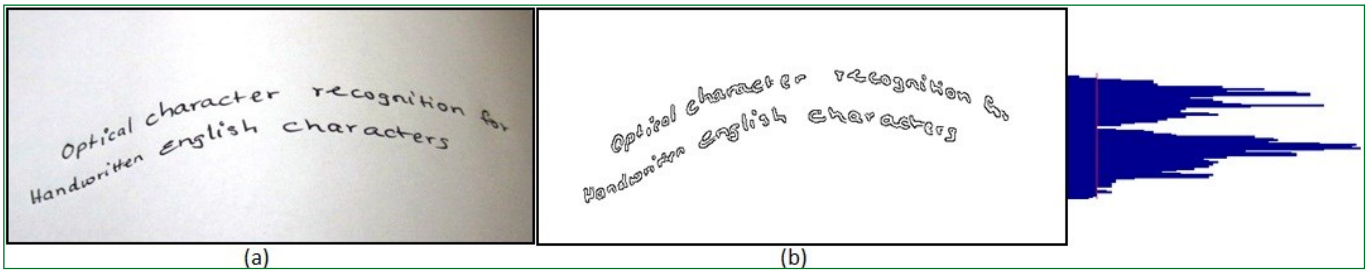

Fig. 1- (a) Original handwritten document image with curled text-line and **(b)** Binarized image with its Projection profile graph

Let T_1 be the set threshold value for text region detection and p_i be the number of pixels in *i*th bin within a block of adjacent rows, then a highly rich text regions should satisfy the [Eq-1].

$$p_i \geq T_1 \quad \forall_i . \tag{1}$$

Text-Line Path Detection

Text-line path detection is the key task in document image analysis where its main goal is the labeling process consisting of assigning the same label to spatially aligned units which include pixels, connected components and or characteristic points. In this case we perform text-line path detection based on connected component which are the individual text characters and are assumed to form text-lines as they are spatially aligned. Based on the nature of our work to extract arbitrary text-lines we adapted the approach presented in [3,2], that use a bank of rotated anisotropic Gaussian filters, the choice to adopt this segmentation technique was arrived due to the complexity of the text-line structure, since our dataset are characterized by multi-oriented, curled/curved text-lines and skewed text and the ability of the algorithm to robustly address these type of problems as they are capable of handling arbitrary text-line orientations due to its rotating anisotropic filter kernel. The use of the rotating smoothing filter by Magnier et al. [26] shows a significant improvement and to a greater degree capable to handle and address the problems associated with any form of text-lines irrespective of their orientation.

The process of filtering the image, involves building a signal $S_i(x, y)$ for every pixel in gray scale image which is a function that corresponds to 360/ΔΘ scan. For precision purpose we have adapted ΔΘ = 45º and i ={1,2,...8} which corresponds to the 8-neighborhood directions to address the high degree of text-line orientation problem. A range of standard deviation scales are considered σ_x and σ_y for x– and y– directions respectively, these scales play an important role in the decision for the presence of a ridge. We perform preliminary work to determine the width and height of every connected component through morphological operation, where we estimated the maximum and minimum width and height of various bounding boxes in the image $(h_{(\max,\min)}, w_{(\max,\min)}) = \max/\min(h_i^{box}, w_i^{box})$ where i = 1,2,….n We then obtained a collection of 8– smoothed images $I_{(s_{i=1,2,..,8})}$ corresponding to each direction with each having maximum response with the range of standard deviation scales. Further a maximum response from a set of the 8– rotated anisotropic Gaussian kernel is chosen for the current pixel and its corresponding ΔΘ, σ_x and σ_y are noted as the representation of a crest line as shown in [Fig-2(a)]. The rotating bank of anisotropic Gaussian filter can be computed as follows:

$$g(x,y,\sigma_x,\sigma_y,\theta)=C.H\left(R_\theta\begin{pmatrix}x\\y\end{pmatrix}\right)e^{-(x,y)R_\theta^{-1}\begin{pmatrix}\frac{1}{2\mu^2} & 0\\ 0 & 2\lambda^2\end{pmatrix}R_\theta\begin{pmatrix}x\\y\end{pmatrix}}. \quad (2)$$

where C is a normalization coefficient, R_Θ a rotation matrix of angle Θ, while σ_x and σ_y the standard-deviations of the filter bank kernel in x and y directions respectively.

We have employed a ridge detection technique presented by Eberly [15] and as used by Bukhari [2], which is the text-line finder to approximate the medial axes from the smoothed image establishing a relationship with its neighbors thus forming a text-line path representative. The resultant smoothed image with maximum response provides an enhanced text-line structure and a good approximation of a symmetrical medial axis (ridge) as shown in [Fig-2(a)] and [Fig-2(b)] respectively. Ridges are formed with the points where the intensity gray level reaches a local maximum in a given direction.

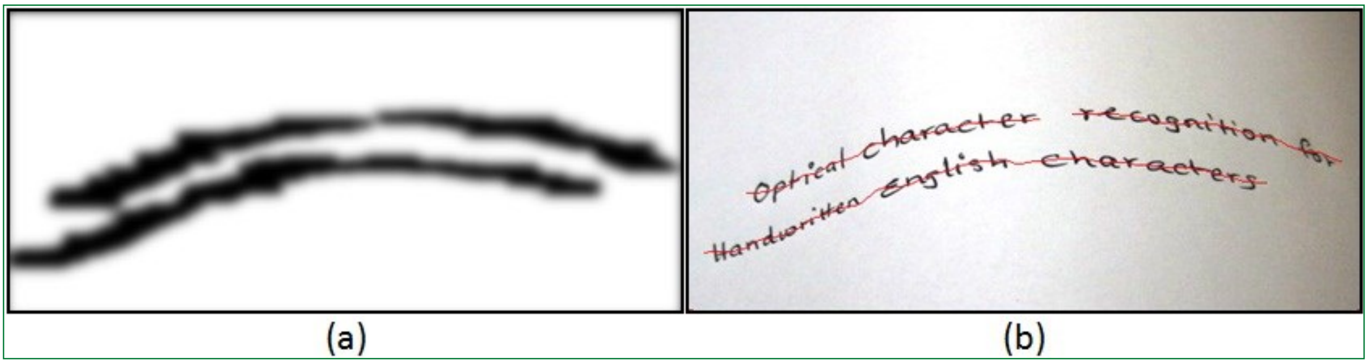

Fig. 2- (a) Anisotropic filter bank results and **(b)** the detected text-line path

Text-Lines Segmentation

Our text-line segmentation combines both the labeling of text-line paths and overlaying it on the binarized image of the original. A label is assigned to each connected component of handwritten document image based on the overlapping ridge. We further employed morphological operation to address the problem where some connected components belonging to the labeled text-line has not been overlapped by the ridge and may have not been regarded to be part of it. The structuring element used depends on the height of the connected components. The labeling process is done according to the number of the unique text-line path, this allows the identification of all the text-lines that are within a specific targeted text regions. Let Tl_i be the number of detected unique text-line paths and l_i be the number of labels associated with each text-line, where i = 1,2,...n Then Tlc_1 is the connected component of the first text-line that are smoothed with the run-length algorithm and is going to be assigned with a crop of colors that visually differentiate individual text-lines.

Text-Line Alignment

Our text-line alignment comprises of the following corrections; skew correction and linear alignment of arc-form text. Vasudev et al [24] gave an arc-form-text to the OCR as input, it recognizes the text region as picture rather than text, whereas the same input if processed into a linear form and fed to OCR there is an improved readability. This demands that OCRs do require linear text for reading. When the resultant text-line segmentation is subjected to OCRs, majority of the OCRs do not read tilted characters because the characters are tilted differently and the efficiency of the OCR becomes relatively low. In order to provide a better input to OCR we have introduce an algorithm that aligns individual arc-form text-strings. The algorithm initially extracts the overall curvature information of an individual text-line and thereafter groups this information into local arcs. Each arc will be processed individually in order to align text strings. Due to the availability of text ascenders and descenders the detected ridges does not produce a smooth curve. To smooth the curve we employed a curve fitting technique to approximate the pixel data by applying a c-means clustering technique to group all the points along the ridge with a constraint that the curve must be smooth with the overall process reducing the amount of pixel points substantially. The cluster points are then used to determine the closeness of one point to its neighbor, and a graph is formed by linking these closest points together using cubic B-spline where the cluster points and links represent vertices and edges respectively. The B-spline is employed to represent the extracted text-line path as they are ideal in handling images with multiple text orientations and curvatures [9] by capturing smoothness inherently present in any text-line layout. With each character's control points are identified and used for fitting the B-spline curve where its order are maintained as 4 to provide curve smoothness that may arise due to ascenders and descenders. The influence of each cluster point is analyzed through its spatial relations by connecting each and every cluster point to its two closest neighbors resulting to a relational graph. The idea is to cluster the data again, with the constraint that the curve passing through the cluster point must be smooth [16].

The next step is to integrate the smoothed text-line path into oriented gradient algorithm in order to extract text-line path gradient orientation and approximate curvature information. The output of this algorithm is the curvature value for each pixel along the text-line path. The output image is then divided according to labeled text-line paths, where curvature information of each is captured by directional gradients to build 1D curvature information.

The supposed 1D curve (from a to b) in the 2D plane is given by the following vector equation:

$$r_t = x_t i + y_t j, \qquad x \le t \le b. \quad (3)$$

Where x_t and y_t are defined and continuously differentiable between t = a and t = b. The t can be thought as time. A point (x_t, y_t) at time t can be approximated as t goes from a to b and we can assume that it never stops, thus allowing us to calculate its speed at all times between a and b.

$$\frac{dt}{dx} = \sqrt{\dot{x}_t^2 + \dot{y}_t^2} > 0. \tag{4}$$

The instantaneous velocity vector (tangent vector) to the curve is estimated as thus,

$$r_t = \dot{x}_t i + \dot{y}_t j. \tag{5}$$

For any value of t the tangent vector r_t makes an angle Θ_t with position x-axis. Thus we can write Θ_t in polar coordinates as,

$$\dot{r}_t = \| \dot{r}_t \| (\cos\theta_t i + \sin\theta_t j). \tag{6}$$

The tangent vector moves along the curve as it rotates in a counter clockwise direction depending whether Θ is increasing or decreasing. While the derivative [Eq-7] below provides information of how fast the curve is turning with its direction ([Fig-3(b)] the length of the green lines). This information provides the curvature and orientation of the curve at points indicated by curve from a to b.

$$\frac{d\theta}{dt}. \tag{7}$$

Thus to compute the curvature of the curve at point (x, y) it is estimated as thus,

$$\tan\theta_t = \frac{\dot{y}_t}{\dot{x}_t}. \tag{8}$$

The same information can be represented in an energy function in [27] as thus,

$$E(\theta_\lambda) = \int \left(a(D\theta_\lambda)^2 + b(DD\theta_\lambda)^2 \right) dt \frac{1}{n}. \tag{9}$$

Where *D* denotes differentiation.

The energy function measures how much the curve is stretched and bent by looking at the differential and curvature at each point of the curve. The first and second terms correspond to the stretching and bending of the curve, which are controlled by the values from *a* to *b*.

We perform transformation of individual text-lines by aligning them horizontally. The techniques involved the extraction of line and distance information. The curved texts are aligned horizontally by transforming curved-form text-line into linear-form based on the curvature information of the curve. The text characters are then extracted and align along the linear text path. Each character's distance to its neighbor is maintained by performing a distance transform using global rotation and translation. Distance transform algorithm finds the pixels values to its neighbor through 8-neighborhood. The alignment step processes each text string individually rendering curved text-lines to be horizontally aligned and transforms it into a form suitable for OCR.

We calculated the length of the text-line path based on two techniques; 1) a general calculation of the curve based on [Eq-10], and 2) calculation of every local curve by modeling an arc [Fig-4]. We then approve the length of the text-line curve by tracking all the points at time t along the text-line path (a to b), this will then align all the pixels a long a straight line ([Fig-3(a)] & [Fig-4], $\overleftarrow{DA}$) with the same length with the curve ([Fig-3], $\overleftarrow{BA}$) and it counts all the number of pixels between the two adjacent character points and later use it to linearly align and order the characters sequentially with the right distance spacing. Calculation of the horizontal text line,

$$H_i = \sum_{t=1tox} (h \times v \times \tan\theta) \geq \max(x). \tag{10}$$

The curvature of the curve allows us to estimate the radius and the origin (centre of the circle). Again from [Fig-3] we can see an identified text path in form of non-uniform and multi-oriented curved lines. At the starting point we estimated the curvature of the self said pixel and adjacent consecutive points along the text path.

$$T_1 \leq p_{c\_val} \leq T_2. \tag{11}$$

where T_1 and T_2 are lower and upper limits set for every point based on a calculated error value.

Fig. 3- (a) Text-line segmentation based on labeling with individual crop of colored labels and **(b)** Text-line segmentation results through allocation of crop of colors to individual text-lines

The next step is to group all adjacent points in the curve that are within the set curvature range [Eq-11], we determined the mean of all the grouped points and determine the median point with all the points in each group used to estimate a local smooth curve (arc) that comprises all the points in the group. [Fig-4] can be taken as a local curve estimated using grouped adjacent points with same curvatures. Our assumption on the curve is that it must be smooth for all the associated pixels and there will be both translation and rotation of image pixels along *x* and y-axis while there will be no depth information (Z value) in the transformation.

From [Fig-4], we can perform the transformation of a point in the curve into a straight line through the following transformation technique. Let A = (x_a, y_a, f)' , B = (x_b, y_b, f)' and r be the radius of the curve with the bend segment subtends an angle Θ at the centre of the circle O. Let point A map to point D, such that D = (x_d, y_d, f)' after line transformation. Then the coordinates of D are given by,

$$x_d = x_a + r(\theta - \sin\theta), \tag{12}$$

$$y_d = y_a + r(1 - \cos\theta), \tag{13}$$

$$z_d = f. \tag{14}$$

Let the point C such that C = (x_c, y_c, f)' be such that $x_b \le x_c \le x_a$, and let C map to point C' such that C' = ($x_{c'}$, $y_{c'}$, f) after line transformation.

Let the angle subtended by the arc AC at the centre O be Φ where,

$$\phi = \frac{x_a + r\theta - x_c}{r} = \theta - \left(\frac{x_c - x_a}{r}\right) \quad (15)$$

Now the coordinates of C' can be calculated as given below,

$$x_{c'} = x_a + r(\phi - \sin\phi), \quad (16)$$

$$y_{c'} = y_a + r(1 - \cos\phi), \quad (17)$$

$$z_d = f. \quad (18)$$

Note that for points of C in the original document with $x_c = x_a$ we have no line transformation.

The modelled of [Fig-4] can estimate every character radius and its corresponding circle centre along the identified curved text path using [Eq-19].

Let $p_2 = (x_t, y_t)$, $p_1 = (x_{t-1}, y_{t-1})$, and $p_3 = (x_{t+1}, y_{t+1})$ be three adjacent points along the text path between points a and b such that $a \le p_1$, p_2, $p_3 \le b$, then radius (r) at time *t* can be computed as follows,

$$r_t = \frac{|p_1 - p_2|}{2\sin\theta_t} \quad (19)$$

Where Θ_t is the angle $\llcorner p_1p_2p_3$.

[Fig-5] shows the ultimate results of our proposed work for text-line alignment.

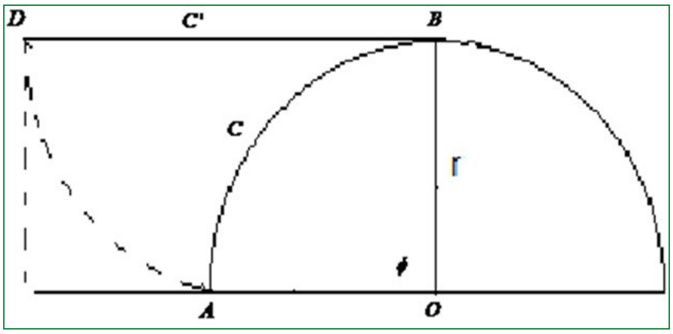

Fig. 4- Diagrammatic representation of local curve alignment

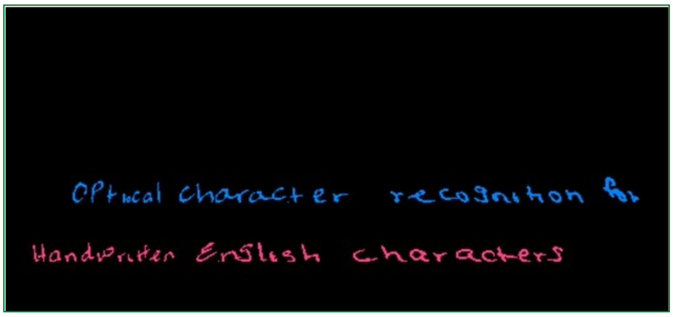

Fig. 5- Horizontally aligned text-string

Experimental Results

Handwritten document images are characterized by multi-oriented or highly skewed text strings, overlapping or curled text strings. It is highly unlikely that the writer purposely writes handwritten documents in a curved form. In this case we have modeled our problem to treat curled-form text-line as curved. To the best of our knowledge there is no standard database available in the literature for curled handwritten documents. We have therefore performed experimentation on both publicly available dataset which are collected from internet including ICDAR dataset and our own video frames of handwritten curled materials captured using a digital camera. Document images in our dataset consist of curl text-lines. Our curled dataset is acquired using digital camera positioned near the top or bottom edges of the document, this allows us to have an image with slightly curled text-line where the text-line path can be estimated in form of a curve and thus the radius and centre of the circle for every local group of arcs can be estimated.

The main advantage of using the digital camera to capture dataset was to obtain high quality curled document images, whereas the important factor which we considered in order to use the camera video frames was the image sharpness. From the results of our experimental metrics to measure image quality, the good results were obtain from video frames recorded by digital camera as opposed to those captured through Iphone or Sumsang [28]. Image sharpness quality is critical in the extraction of text characters as they give better results due to its high contrast. Our dataset contains 252 grayscale and binarized document video frames of pages captured from several technical books and planar sheets from different writers captured by an off-the-shelf handheld digital camera in a normal classroom environment.

The alignment process in a linear and sequential manner allows for the calculation of curvature across each character. In order for the algorithm produce favorable results we first extracted an array of points of each character and normalize them to reduce their dimension without missing their relevance from the image. Though projection profiles has been adapted for handwritten documents with little overlap for text line segmentation as in [6], we employed it for the identification of highly overlap text regions with the benefit of reducing computational space and time. Adjacent line connection between lines is very frequent due to ascenders and descenders and some errors may occur.

Handwritten characters unlike printed characters suffers with character-character spacing, as there is a variation in character wise spacing in handwritten documents, during smoothing process it makes hard to obtain a uniform spacing, this we opted to use a much larger width area and standard deviation along x-direction in order to try to compensate it, we also maintain a low degree of binary image conversion threshold. The technique is robust as it is build to perform segmentation of text-line and alignment at character level which significantly reduces the computational time. The overlapping text-lines can be segmented based on the smoothed image by finding the local minimum on the portion that touches each other.

Evaluation Metrics

We evaluated the text-line segmentation based on metrics for the performance measure of our algorithm according to text-line segmentation accuracy in comparison to the ground truth. The ground truth generated through manual annotation. Our proposed technique result reveals that there is an improvement as compared to existing methods. Following the text-line segmentation, the text-line alignment though it's an indispensable pre-processing step in the analysis of handwritten document, its performance was not measured, but visually it appears to be good. The above metrics was introduced to test and provide a platform level that will significantly portray workability of our proposed method with any public OCR system.

For line segmentation [Table-1], the rate reaches 98.20% for linear-form text-lines, whereas for curled, curved or arc-form text-lines reaches 91.73%. This accuracy is defined by (number of text lines detected / number of text lines of the document) X 100 [24]. In our metrics we considered three types of errors: 1) split lines, 2) joined lines and 3) lines with out-lier components. Split lines is wrongly divided text-lines, joined lines, joined lines corresponds a sequence of short overlapping lines and was not uniquely segmented and lines with out-lier components corresponds to lines that contains other components that is not part of the perceived text-line and this include character components or words from other text-lines and are adjacent to the text-line under consideration.

Table 1- Evaluation metric

Technique	Total number of lines				Percentage of lines (%)			
	Existing [24]		Proposed		Existing [24]		proposed	
	Curve	linear	curve	linear	curve	linear	curve	linear
Correctly segmented	282	627	366	655	70.68	94	91.73	98.2
Split lines	67	7	17	0	16.79	1.05	4.26	0
Joined lines	32	12	8	5	8.02	1.8	2.01	0.75
Lines including out-lier words	18	21	8	7	4.51	3.15	2.01	1.05

We have considered the error rate of 0.5% that is caused by the presence of noise. The noise comes from different sources ranging from light illumination, effectiveness of acquisition devices and the quality of the document material based on age or texture. The effectiveness of the algorithm is illustrated on a sample of 252 documents chosen randomly among two datasets of documents, from each dataset. To identify and visualization purpose individual lines, each line and the consecutive lines is presented in different colors.

Experimentation on Public Dataset

To test the performance of our algorithm, real-world handwritten documents were collected from the various writers including ICDAR dataset, most of the document only required global skew correction while some had varied skew-angle and required to be corrected locally [Fig-6]. The results of our algorithm show an improved result on the handwritten documents as it performs text-line alignment locally based on the local curvature and orientation information.

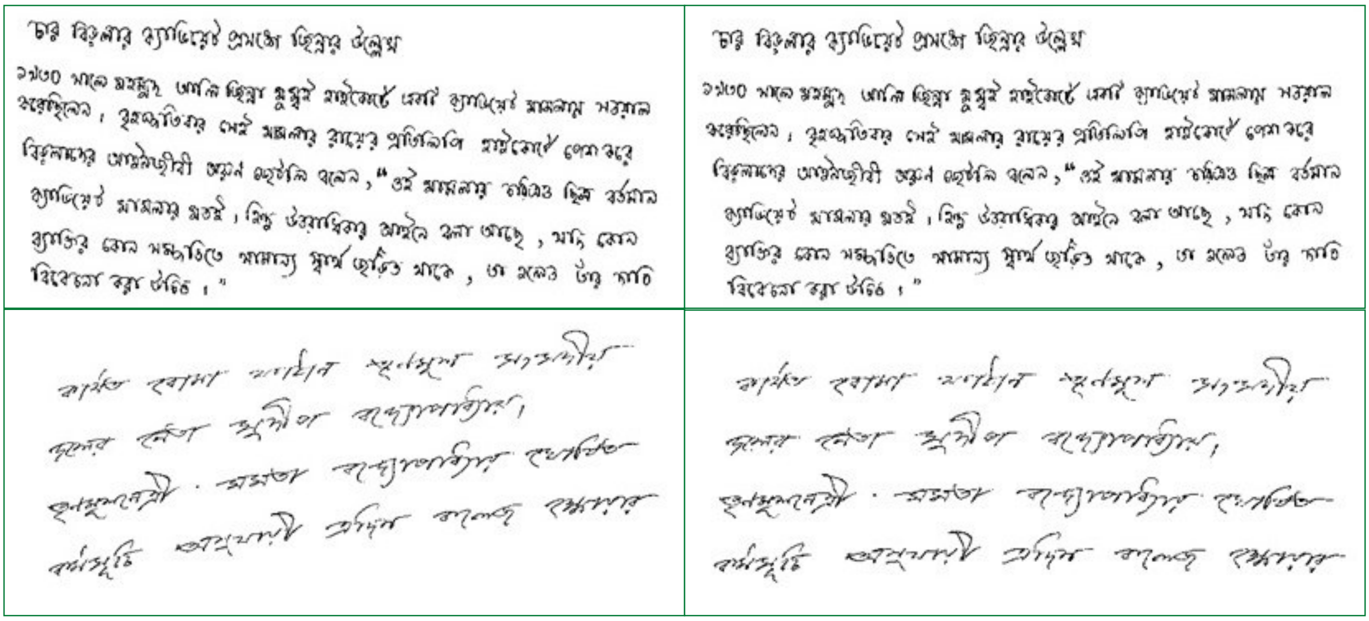

Fig. 6- First column- Original images and Second column- rectified images

Experimentation on Our Own Dataset

While collecting our dataset we tried to make writers provide text-lines that are multi-oriented have different degrees of orientations the same document. [Fig-7] shows one of our sample document text-line paths, with its corresponding smoothed line using B-spline and the curvature information. The longer the bin [Fig-7(c)] (green colored) represent more curvature. The orientation is also evident as the text-line path turns. [Fig-8] shows more experiments

Experimentation on Text-lines Bifurcations (Branching)

The individuality of text-line structure is exclusively determined from the relationship among the local physical aligned characteristics. Therefore, the extracted text-line pattern is firstly used to locate the key points that are relatively stable, unique and repeatable. From each labeled connected components representing the text-lines we determine text-line bifurcation and endings points, these are then selected as key points to extract local text-line properties. A text-line bifurcation is defined as the text-line point where there is diverges into branch text-lines, and the text-line ending is the point at which the text-line ends or disappears abruptly. This disappearance could be due to the abrupt ending of text-lines at the page margins or their poor visibility from the imaging system. In order to extract the text-line endings and bifurcation points we examining the connectivity of every pixel and determine the crossing number [29] for every pixel. The crossing number is the sum of differences between pairs of adjacent pixels in 3X3 window centered at *m* and is calculated as thus,

$$R(m) = \sum_{z=1}^{8} | val(m_{z \bmod 8}) - val(m_{z-1}) |. \quad (20)$$

The pixel m with val(m) = 1 corresponds to text-line ending point if R(m) = 2, and corresponds to text-line bifurcation point if R(m) ≥ 6. [Fig-9], shows sample experiment on text branching.

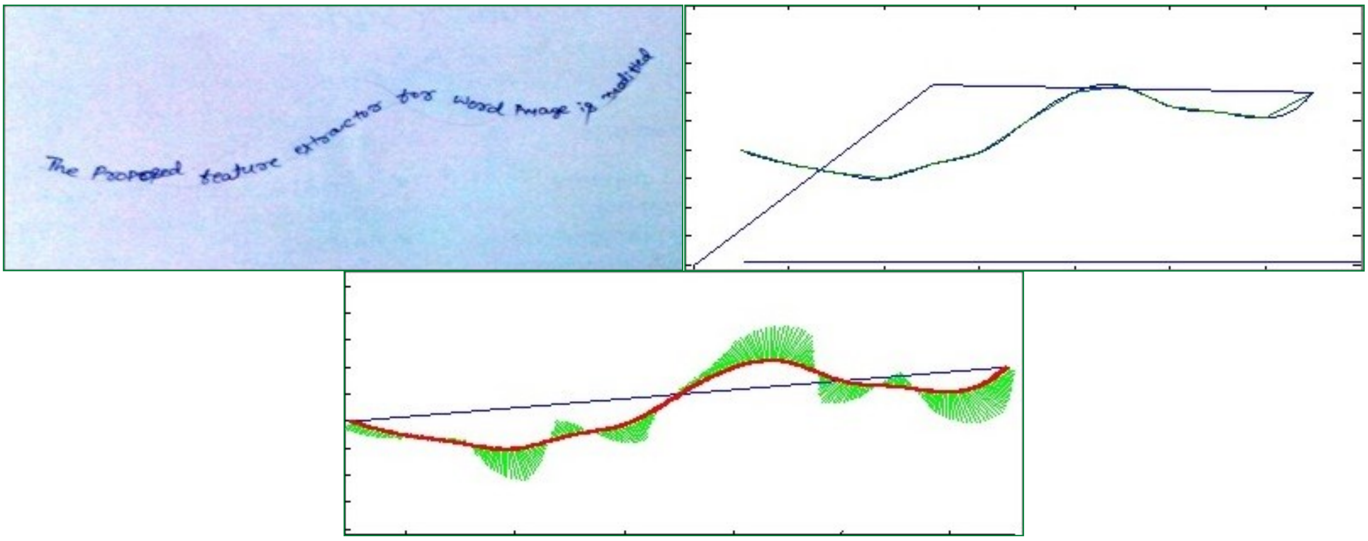

Fig. 7- (a) Original multi-oriented curved document image, **(b)** Text-line path with B-spline and **(c)** Curvature features

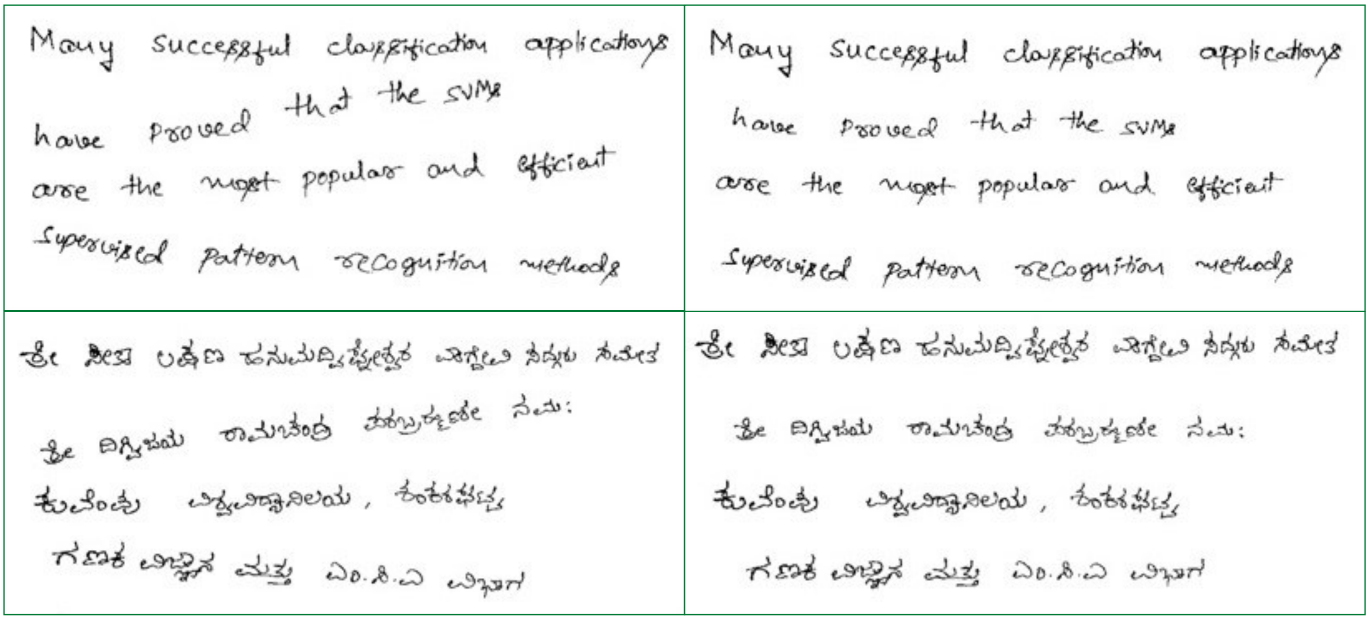

Fig. 8- First column: Original images and Second column: rectified images

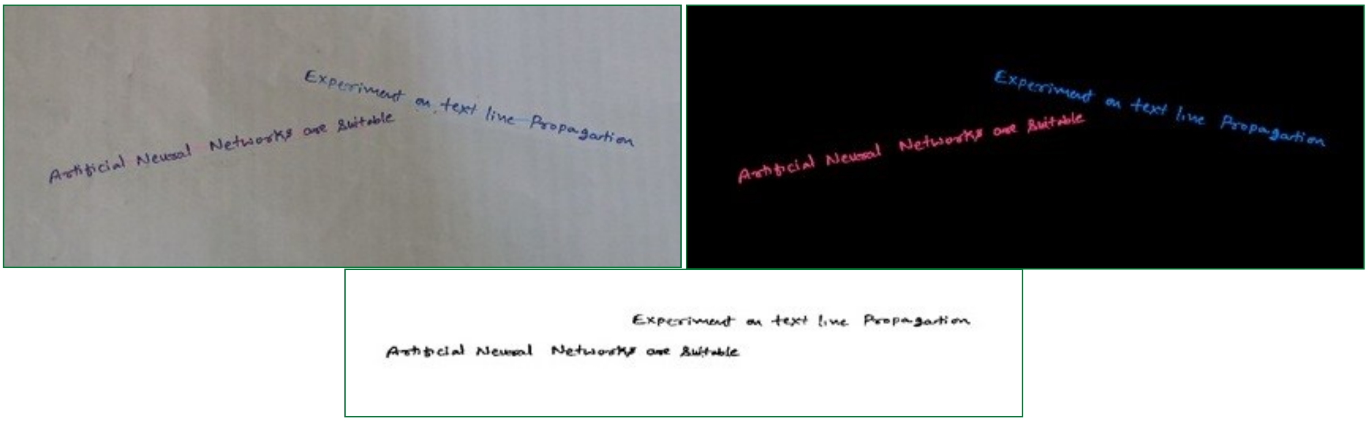

Fig. 9- (a) Original branch text-line image, **(b)** segmented text-line and **(c)** aligned text-line image

Conclusions

In this paper we present a novel approach to build an algorithm that provides OCR-ready handwritten documents for recognition. The arbitrary handwritten text-lines in document images are segmented based on their spatial alignment of connected components through finding of medial axis of the smoothed image. Oriented gradients and curvature features of identified text-line path are estimated. Finally the estimated individual text-line orientation and curvature information are used to horizontally align all the document text strings. Our method overcomes most of these limitations and since it involves only character rotation, it is free from character deformation resulting to a better candidate text for OCR. The main benefit of our method is that it can handle the presence of multi-oriented text-lines within the same image.

Conflicts of Interest: None declared.

References

[1] Arica N. & Yarman-Vural F.T. (2001) *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 31(2), 216-233

[2] Bukhari S.S., Shafait F. & Breuel T.M. (2009) *IEEE 10th International Conference on Document Analysis and Recognition,* 446-450.

[3] Bukhari S.S., Shafait F. & Breuel T.M. (2011) *IEEE International Conference on Document Analysis and Recognition,* 579-583.

[4] Ulges A., Lampert C.H. & Breuel T.M. (2005) *IEEE 8th International Conference on Document Analysis and Recognition,* 1001 -1005.

[5] Yan H. (2001) *IEEE 6th International Conference on Document Analysis and Recognition,* 266-270.

[6] Likforman-Sulem L., Zahour A. & Taconet B. (2007) *International Journal of Document Analysis and Recognition*, 9(2-4), 123-138.

[7] Nobile N. & Suen C.Y. (2014) *Handbook of Document Image Processing and Recognition*, 257-290.

[8] Li Y., Zheng Y., Doermann D. & Jaeger S. (2008) *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(8), 1313-1329.

[9] Kasar T. & Ramakrishnan A.G. (2013) *International Journal of Computer Vision and Signal Processing*, 3(1), 1-9.

[10] Panwar S. & Nain N. (2013) *IEEE 4th National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics,* 1-4.

[11] Antonacopoulos A. & Ritchings R.T. (1994) *IEEE 12th International Conference on Pattern Recognition, 2,* 339-344.

[12] Cohen E., Hull J.J. & SriHari S.N. (1991) *International Journal of Pattern Recognition and Artificial Intelligence*, 5(1-2), 221-264.

[13] Cao H. & Govindaraju V. (2009) *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(7), 1184-1194.

[14] Kurniawan F., Rehman A. & Mohamad D. (2009) *IEEE International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering,* 1-4.

[15] Eberly D., Gardner R., Morse B., Pizer S. & Scharlach C. (1994) *Journal of Mathematical Imaging and Vision*, 4(4), 353-373.

[16] Gavali V. & Bombade B.R. (2013) *International Journal of Computer Science and Mobile Computing*, 2(6), 285-293.

[17] Ouwayed N. & Belaïd A. (2012) *International Journal on Document Analysis and Recognition*, 15(4), 297-314.

[18] Belaïd A. & Ouwayed N. (2012) *Guide to OCR for Arabic Scripts,* 103-122.

[19] Cruz F. & Ramos T.O. (2013) *IEEE 12th International Conference on Document Analysis and Recognition,* 718-722

[20] Yin F. & Liu C.L. (2007) *IEEE International Conference on Wavelet Analysis and Pattern Recognition*, 3, 1123-1128.

[21] Yin F. & Liu C.L. (2008) *International Conference on Frontiers in Handwriting Recognition*, 229-234.

[22] Pal U. & Roy P.P. (2004) *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, 34(4), 1676-1684.

[23] Chiang Y.Y. & Knoblock C.A. (2011) *IEEE International Conference on Document Analysis and Recognition*, 1399-1403.

[24] Vasudev T., Hemanthkumar G. & Nagabhushan P. (2007) *Pattern Recognition Letters*, 28(16), 2343-2351.

[25] Manmatha R. & Srimal N. (1999) *2nd International Conference on Scale-Space Theories in Computer Vision,* 22-33.

[26] Magnier B., Montesinos P. & Diep D. (2011) *Advanced Concepts for Intelligent Vision Systems,* 261-272.

[27] Takahashi S. (1998) *The Visual Computer*, 14(5), 208-227.

[28] Boaz T.K. & Prabhakar C.J. (2014) *IEEE International Conference on Contemporary Computing and Informatics*, 503-508.

[29] Chaudhuri S., Chatterjee S., Katz N., Nelson M. & Goldbaum M. (1989) *IEEE Transactions on Medical Imaging*, 8(3), 263-269.