EURASIP Journal on Image and Video Processing
a SpringerOpen Journal

RESEARCH Open Access

A novel method for straightening curved text-lines in stylistic documents

Brij Mohan Singh[1*], Ankush Mittal[2] and Debashis Ghosh[3]

Abstract

Stylistic text can be found on sign boards, street and organizations boards and logos, bulletin boards, announcements, advertisements, dangerous goods plates, warning notices, etc. In stylistic text images, text-lines within an image may have different orientations such as curved in shape or not be parallel to each other. As a result, extraction and subsequent recognition of individual text-lines and words in such images is a difficult task. In this paper, we propose a novel scheme for straightening of curved text-lines using the concept of dilation, flood-fill, robust thinning, and B-spline curve-based fitting. In the proposed scheme, at first, dilation is applied on individual text-lines to cover the area within a certain boundary. Next, thinning is applied to get the path of the text, approximate the path using the B-spline, find the angle between the normal at a point on the curve and the vertical line, and finally visit each point on the text and rotate by their corresponding angles. The proposed methodology is tested on variety of text images containing text-lines in Devanagari, English, and Chinese scripts which is evaluated on the basis of visual perception and the mean square error (MSE) calculation. MSE is calculated by line fitting applied on input and output images. On the basis of evaluation results obtained in our experiments, the proposed method is promising.

Keywords: OCR; Dilation; B-spline; Flood-fill; Feature extraction; Recognition; Segmentation; Thinning

I. Introduction

A large volume of research effort has been dedicated to OCR systems. Numbers of algorithms [1-6] are available for this purpose, and many commercial OCR systems [7,8] are now available in the market but most of these systems can recognize only text images having straight text-lines (horizontal) and designed only for a specific script or language. On the other hand, there are few limitations in regard to the source materials and character formatting which make feature extraction and recognition difficult.

Increasing demand for stylistic text recognition attracts researchers for the designing and development of new algorithms that can handle such type of text due to usage in many potential applications including container identification mark recognition system [9], vehicle license plate recognition [10], text recognition in video and images [11-14], image retrieval [15] from the database, intelligent transport systems [16], robotics [17], and text translation service for tourist assistance [18], for foreigners with language barrier, etc. Stylistics text as shown in Figure 1 can be found on greeting cards, front cover of books, organization's logos, stamp seals, maps, engineering drawings, advertisements, newspaper's front page, hallmark cards, signboards, hording boards, archives, etc., containing multi-oriented, multi-font size, and curved text-lines or words resulting in errors in the recognition process. There may be two ways for the recognition of such documents: one is to develop algorithms those can recognize words and text-lines in actual format (curved etc.), and second is to convert curved text-lines into straight text-lines, preferably oriented horizontally. Further, the straight text-lines must be segmented into words or characters before going for recognition process. The main advantage of proposed approach is that no specific feature extraction and classification techniques or dataset is required in the recognition of such documents. So, in view of this, text normalization plays important role in the recognition of stylistic text.

II. Related works

Many pieces of works are available on the document image recognition [2,5,6] having straight text-lines and words. But in the literature, there are only few works available towards the recognition of stylistic documents [19-44].

* Correspondence: bmsingh1981@gmail.com
[1]Department of Computer Science and Engineering, College of Engineering Roorkee, Roorkee 247667, India
Full list of author information is available at the end of the article

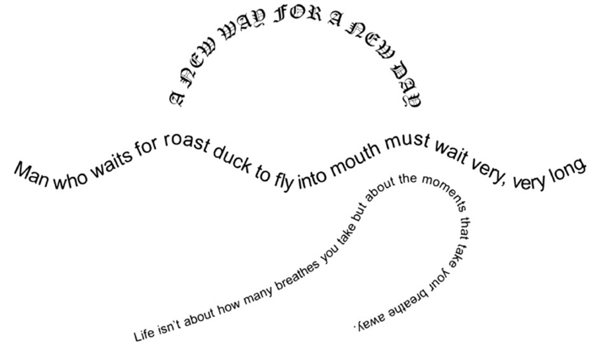

A NEW WAY FOR A NEW DAY
Man who waits for roast duck to fly into mouth must wait very, very long
Life isn't about how many breathes you take but about the moments that take your breathe away.

Figure 1 **Sample of curved text-lines.**

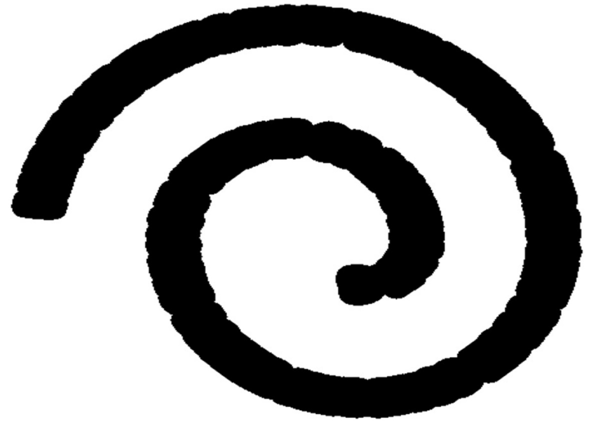


Figure 3 **The result after dilation using a circular structuring element of radius 17.**

In 1991, Xie et al. [19] proposed a pattern recognition system invariant of translation, scale change, and rotation transformation of pattern and 97% recognition accuracy is obtained from the 10 Arabic numerals. Another work on English stylistic text recognition due to Tang et al. [20] used a translation-ring-projection algorithm to handle the multi-oriented English alphabets in 1991.

In 2000, one work on English stylistic text recognition is due to Adam et al. [21] in which an approach of recognition of multi-oriented and multi-scaled character in engineering drawings is proposed. Fourier-Mellin transform is used to recognize the characters. This approach is limited to the recognition of few characters and it is time consuming. Further in 2001, Yang et al. [22] proposed an approach of three stages for multi-oriented Chinese character recognition where features are mainly based on geometric measures of the foreground pixels of the characters. It is limited to Chinese language only.

In 2003, Hase et al. [23] proposed a multi-oriented character handling approach based on the character types such as inclined, horizontal, vertical, curved, etc., and it considers character realignment horizontally and then for recognition. The main drawback of Hase et al. [23] approach is the distortion due to realignment of curved text. For rotated and/or inclined English character recognition, Hase et al. [23] used a parametric Eigen-space-based approach. This method is also limited to English character recognition but not used in variation of font style, size, and multi-script.

In 2005, Pal et al. [24] proposed a recognition-based approach to handle Indian multi-oriented and curved text. It is based on the water reservoir concept for segmentation of characters from stylistic documents without any skew correction. Next, individual characters are recognized. This approach is limited to recognize Bangla and Devanagari script text only. In 2006, Hayashi et al. [25] proposed a rotation invariant Arabic numerals recognition system where a numeral is divided into elementary sub-patterns like straight line, C-shaped line, and 0-shaped line using thinning algorithm, and then recognized based on different features like curvature, angle information, length, arc-length, etc. of the sub-pattern. In 2006, Pal et al. [26] proposed a method towards the recognition of multi-oriented and multi-sized English characters based on the modified quadratic discriminate function (MQDF). The main drawback of the proposed approach is that it cannot distinguish similar looking character such as 'b' and 'q', 'p' and 'd', 'n' and 'u', etc. This is because of the use of rotational invariant features.

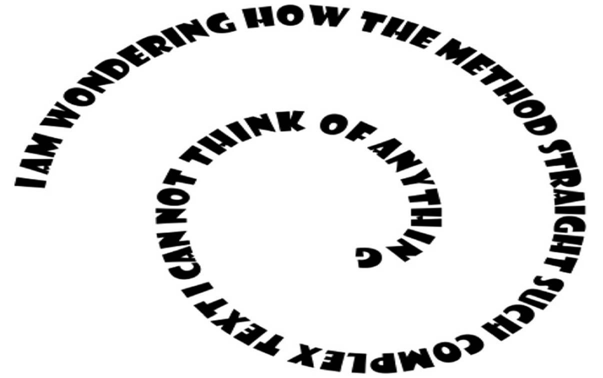

I AM WONDERING HOW THE METHOD STRAIGHT SUCH COMPLEX TEXT I CAN NOT THINK OF ANYTHING

Figure 2 **Curved text-line.**

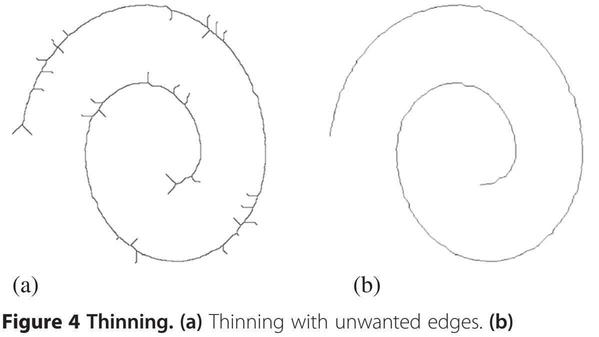


Figure 4 **Thinning.** **(a)** Thinning with unwanted edges. **(b)** Smoothed thinning.

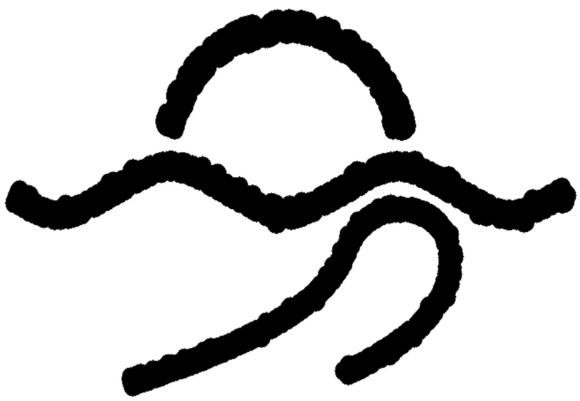


Figure 5 Dilation on the text-lines.

In 2007, Monwar et al. [27] proposed an approach of recognizing printed alpha-numeric character of different angle and in this approach, each character is described by a small set of two-dimensional characteristic views of different angles for feature extraction. In 2008, Roy et al. [28] proposed an approach towards recognition of English character in graphical documents containing multi-scale and multi-oriented text. For recognition of such multi-scale and multi-oriented characters, a support vector machine (SVM)-based scheme is presented.

In 2010, Pal et al. [29] proposed an approach for the recognition of multi-oriented Bangla and Devnagari characters. Although it is also a recognition-based system, this system fails in confusing characters of Bangla and Devnagari script.

In 2011, Chiang et al. [30] proposed a general text recognition technique to handle non-homogeneous text by exploiting dynamic character grouping criteria based on the character sizes and maximum desired string curvature. In 2011, Shivakumara et al. [31] proposed an approach to detect multi-oriented text in videos. The input image is first filtered with Fourier-Laplacian, and K-means clustering is then used to identify candidate text regions based on the maximum difference. The skeleton of each connected component helps to separate the different text strings from each other. Finally, text string straightness and edge density are used for false-positive elimination. Shivakumara et al. [31] method is limited to English language text orientation.

All the approaches discussed earlier based on the recognition of multi-oriented characters and are limited to proper script recognition. There is not a single approach in the literature which is based on the straightening of curved text-lines or words and script independence. In contrast, an approach for curved text-line straightening is proposed in this work which can handle multi-font size and type and multi-script text-lines in a single document.

III. Proposed approach

In this work, the images to be processed are captured by a scanner and camera. For the thresholding of text images, Gatos [32] method is used. Figure 2 shows a sample of curved text-line image.

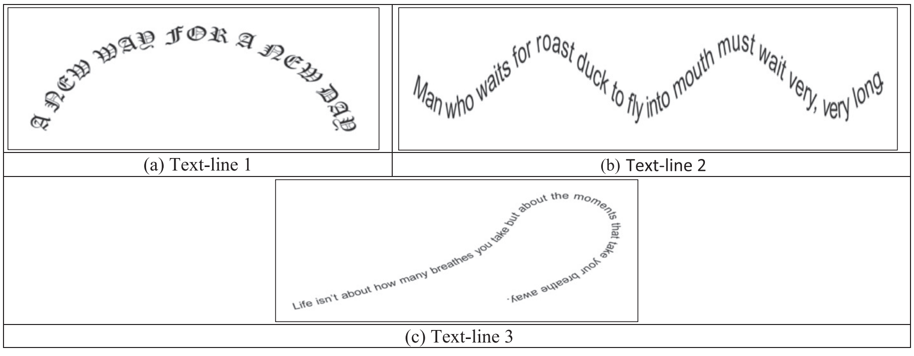
Curved or stylistic text present in document images poses problems in segmentation and recognition. So, before recognition, it is important to straighten the text-lines. This paper attempts to present a method, which is based on the following important steps:

Step 1

Apply morphological dilation operation [33] on the text document using a disk-based structuring element with radius 17 so that region grows in the form of circular bubbles until all foreground objects (black pixels) are covered within a certain boundary. The selection of radius size of structuring element is a manual parameter in our work. It can vary from images to images, but on the basis of experiments, 17 is the most appropriate. Figure 3 shows the curved text image after dilation operation.

Step 2

For the approximation of black region of the curved text-line, we need pixels in a certain order that follows a path coinciding with the curve. This is accomplished by applying thinning to convert the region to one-pixel-thick lines.



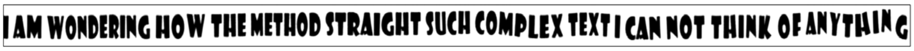
A NEW WAY FOR A NEW DAY
Man who waits for roast duck to fly into mouth must wait very, very long
Life isn't about how many breathes you take but about the moments that take your breathe away.

(a) Text-line 1

(b) Text-line 2

(c) Text-line 3

Figure 6 Segmented English text-lines with different font styles. (a) Text-line 1. **(b)** Text-line 1. **(c)** Text-line 3.



I AM WONDERING HOW THE METHOD STRAIGHT SUCH COMPLEX TEXT I CAN NOT THINK OF ANYTHING

Figure 7 Final output of curved text-line straightening approach.

Since general thinning method results in many unwanted edges along the curve, as shown in Figure 4a, thinning method given in [34] is applied to overcome the shortcomings. Figure 4b shows the result of the smoothed thinning. Thinning is based on the following steps:

- The image is inverted, '1' as black and '0' as white for the MATLAB perspective.
- Erosion is applied on dilated image using disk-based structuring element of size '17'.
- Morphological operation of thinning is applied on eroded image with '10' iterations.
- The output image is again inverted for the better working of further steps of curve straightening.

Step 3

In this step, curve fitting is applied on the result obtained from the previous step. For the curve fitting, B-spline [35-41] curve is used instead of polynomial curve to approximate the pixel data. Polynomial curve fails to approximate complex curve as shown in Figure 2. B-spline and Bezier curves have very similar form but Bezier curve contains more information. Approximating simple curve using polynomial curve suffers with Runge's phenomenon [42-44]. So, when we approximate the higher-degree pixel data, the accuracy does not always increase. Hence, in the proposed approach, B-spline curve is used, which is free from Runge's phenomenon even at higher degrees. B-spline curve is described as follows:

Given $n + 1$ control points P_0, P_1,........., P_n and a knot vector $U = \{u_0, u_1, \ldots\ldots\ldots u_m\}$, the B-spline curve of degree p defined by these control points and knot vector u is

$$C(u) = \sum\nolimits_{i=0}^{n} N_{i,p}(u) P_i \quad (1)$$

Where $N_{i,p}(u)$s are B-spline basis functions of degree p.

The degree of B-spline basis functions is defined by p. The ith B-spline basis function of degree p, written as $N_{i,p}(u)$, is defined recursively as follows:

$$N_{i,0}(u) = \begin{cases} 1 & \text{if} \quad u_i \le u < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \tag{3}$$

If we have $n + 1$ data points D_0, D_2,..........., D_n and want to find a B-Spline curve that can follow the shape of the data polygon without actually containing the data points, then there is need of two more inputs. The first input is the number of control points and second input is degree (p), where $n > h \ge p \ge 1$. With these two inputs, a set of parameter and a knot vector can be determined. Let the parameter be t_0, t_1,..., t_n. The numbers of data points are equal to the number of parameters. Now, the approximation B-spline of degree p is given by:

$$C(u) = \sum\nolimits_{i=0}^{h} N_{i,p}(u) P_i \tag{4}$$

Where P_0, P_1, ..., P_h are the $h + 1$ unknown control points.

After passing the first and last data points to curve, $D_0 = C(0) = P_0$ and $D_n = C(1) = P_h$, there are only $h - 1$ unknown control points. Taking this into consideration, the curve equation becomes the following:

$$C(u) = N_{0,p}(u) D_0 + \left(\sum_{i}^{h-1} N_{i,p}(u) P_i \right) + N_{h,p}(u) D_n \tag{5}$$

To find out control points P_1,... P_{h-1} such that the function $f\,(P_1,\ldots,P_{h-1})$ is minimized. The approximation is done using least square method. The sum of all squared distances is

$$f(P_1, \ldots\ldots, P_{h-1}) = \sum\nolimits_{k=1}^{n-1} |D_k - C(t_k)|^2 \quad (6)$$

$$\sum_{k=1}^{n-1} N_{g,p}(t_k) \sum_{i=1}^{h-1} N_{i,p}(t_k) P_i = \sum_{k=1}^{n-1} N_{g,p}(t_k) Q_k \quad (7)$$

Since we have $h - 1$ variables, g runs from 1 to $h - 1$ and there are $h - 1$ such equations:

$$P = \begin{bmatrix} P_1 \\ P_2 \\ . \\ . \\ . \\ P_{h-1} \end{bmatrix} \tag{8}$$

(a) (b)

Figure 8 Sample with jaggies and after applying anti-aliasing. (a) Output with jaggies. **(b)** Smoothed edges after applying anti-aliasing.

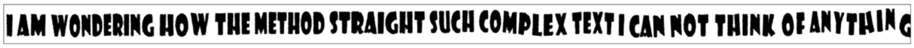

I AM WONDERING HOW THE METHOD STRAIGHT SUCH COMPLEX TEXT I CAN NOT THINK OF ANYTHIN G

Figure 9 Smoothed straight text-line.

$$Q = \begin{bmatrix} \sum_{k=1}^{n-1} N_{1,p}(t_k)Q_k \\ \sum_{k=1}^{n-1} N_{2,p}(t_k)Q_k \\ \cdot \\ \cdot \\ \cdot \\ \sum_{k=1}^{n-1} \mathrm{N}_{h-1,p}(t_k)Q_k \end{bmatrix} \quad (9)$$

$$N = \begin{bmatrix} N_{1,P(t_1)} & N_{2,p}(t_1)\cdots & N_{h-1,p}(t_1) \\ N_{1,p}(t_2) & N_{2,p(t_2)}\cdots & N_{h-1,p}(t_2) \\ \vdots & \vdots & \vdots \\ N_{1,p}(t_{n-1}) & N_{2,p}(t_{n-1}) & N_{h-1,p}(t_{n-1}) \end{bmatrix} \quad (10)$$

The system of linear equation can be rewritten as

$$\left(N^T N\right)P = Q \quad (11)$$

$$P = \left(N^T N\right)^{-1} Q \quad (12)$$

Since N and Q are known, solving this system of linear equations for P gives the desired control points.

Step 4

Stylistic documents may have more than one curved or multi-oriented text-lines as shown in Figure 1. Before the straightening of text-lines, it is important to extract individual text-lines first. Multi-text-lines segmentation is dealt in the following manner:

A. Dilation is applied on each of the text-line to cover all the foreground pixels within the black boundary. Figure 5 shows the dilation on the curved multi-text-lines using a circular structuring element of radius 17. The size of structuring element is chosen on the basis of experiments performed in this work.
B. Modified flood fill algorithm is applied to find out the number of different text-lines in the image and to separate them from the input image. The steps 1 to 4 mentioned earlier are applied for the straightening of the image obtained after segmentation process. Modified flood fill algorithm marks the pixels of a region such that all the pixels of same region got the same numbering. It makes a blank image for each region and copies all the pixels of image (original undiluted image) to the corresponding image of the regions. Hence, separate images are obtained for separate text-lines. The segmented images are shown in Figure 6.

However, segmentation step produces the following errors in the extraction of individual text-line from multi-line images due to the following problems arising during the morphological operations such as:

- If the inter-character spacing is large in stylistic text-lines. Although, it is assumed that the text image to be processed is based on the isolated character recognition. So, if inter-character spacing is large, the recognition rate will not be more affected.
- When two or more text-lines are very close to each other, it will result in merged text-lines. In our experiments, we have taken only those images which contain sufficient gap between text-lines due to the limitation of morphology.

The following pseudo code outlines the modified flood fill algorithm.

```
function floodfill (x,y, endcolor, newcolor)
    if(getpixel(x,y)=oldcolor)
         return
      setpixel (x, y, newcolor)
         floodfill(x-1, y, old color, newcolor)
         floodfill(x+1, y, oldcolor, newcolor)
         floodfill(x, y-1, oldcolor, newcolor)
         floodfill(x, y+1, oldcolor, newcolor)
    end function
Subroutine main ()
           black=0  // foreground
           white=1 // background
           region=2
              for y=1 to image height
                for x=1 to image width
                   if (getpixel (x,y)==black)
                      floodfill (x,y,black, region)
                      region=region+1
                   end if
              end for
                end for
           End subroutine
```

In this step, the least square method [45] is used to approximate the pixel data into B-spline curve. Due to the discrete domain of the sample space, the output

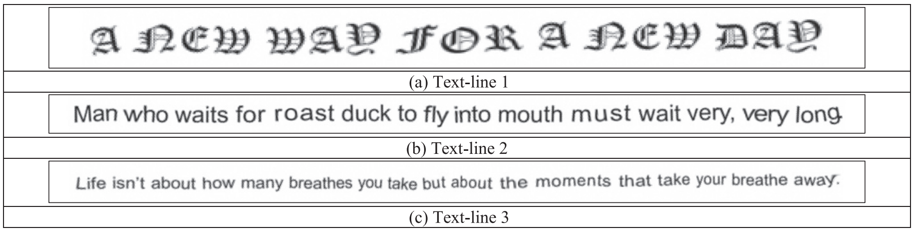

A NEW WAY FOR A NEW DAY
(a) Text-line 1
Man who waits for roast duck to fly into mouth must wait very, very long.
(b) Text-line 2
Life isn't about how many breathes you take but about the moments that take your breathe away.
(c) Text-line 3

Figure 10 The straightened text-lines. (a) Text-line 1. **(b)** Text-line 1. **(c)** Text-line 3.

image may suffer from aliasing effects as shown in Figures 7 and 8a. This will cause serrated character edges in the output image as shown in Figure 8a. To remove the artifact produced by aliasing, anti-aliased pixels are used with bilinear interpolation [33] as shown in Figure 8b. Figure 9 shows the final smoothed output of the proposed approach of straightening of the curved text-line.

Steps 1 to 4 are applied on the text-lines obtained after modified flood fill technique, and the results obtained are shown in the Figure 10.

IV. Experimental results and discussions

The proposed straightening algorithm is tested on 140 images having variety of stylistic multi-oriented text collected from newspapers, books, notebooks, journal articles, magazines, maps, engineering drawings, hoarding, and sign and notice boards. Some images of dataset contain other information along with the text. So, in view of this, we separated multi-oriented text manually from other objects present in the images with the help of MS Paint as shown in Figure 11.

This method is evaluated on the basis of visual perception and the mean square error calculation. Mean square error is calculated by line fitting applied on input and output images with the help of following equations:

The least square line uses a straight line

$$Y = a + bx \tag{13}$$

to approximate the given set of data (x_1, y_1), (x_2, y_2), ….., (x_n, y_n), where $n \geq 2$.

The best fitting curve $f(x)$ has the least square error, i.e.,

$$= \sum_{i=1}^{n} [y_i - f(x_i)]^2 = \sum_{i=1}^{n} [y_i - (a + bx_i)]^2 = \min \tag{14}$$

From Table 1, it is observed that 99.89% error removed from Figure 2; a Devanagari text-line document image; 97.78%, 94.00%, and 99.65% error removed from Figure 6a, 6b, and 6c, respectively, with different font style English text-lines document images; 98.19% error removed from Figure 12b; a Devanagari text-line document image and 99.78% error removed from Figure 12c; English text-line document image and 99.82% error removed from Figure 12d; and Chinese text-line document image.

The proposed method was tested on handwritten as well as machine-printed Devanagari, English, and few

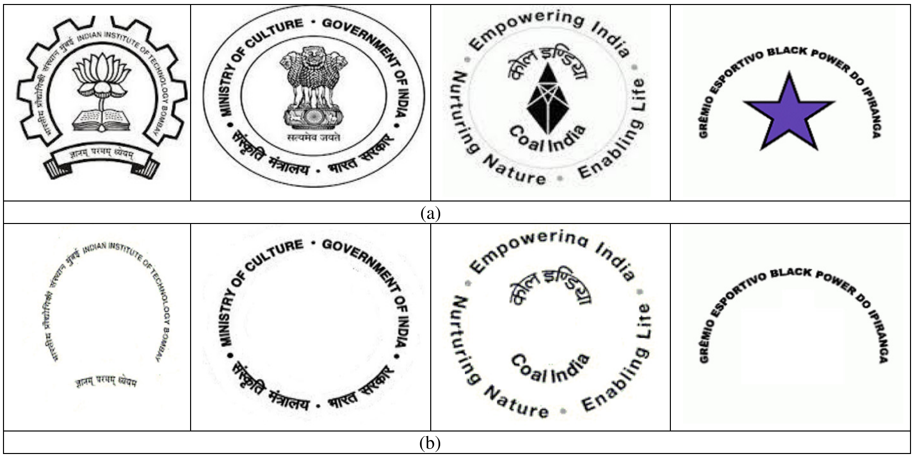

INDIAN INSTITUTE OF TECHNOLOGY BOMBAY
MINISTRY OF CULTURE • GOVERNMENT OF INDIA
Empowering India
Nurturing Nature • Enabling Life
Coal India
GRÊMIO ESPORTIVO BLACK POWER DO IPIRANGA
(a)
(b)

Figure 11 Sample images of dataset used. a) Original images of dataset. **(b)** Manually cropped images used in actual experiments.

Table 1 Evaluation results with respect to used measure

Document image (bmp)	Line fitting error		Percentage error removed
	Before	After	
Figure 2	356,674,630.1	1,108,113.946	99.89
Figure 6a	2,698,347.149	59,830.05687	97.78
Figure 6b	1,182,706.997	70,918.12767	94.00
Figure 6c	4,832,503.452	16,977.00993	99.65
Figure 12a	3,815,193.993	69,062.19421	98.19
Figure 12b	26,762,503.43	58,716.83767	99.78
Figure 12c	16,432,153.41	29,069.77283	99.82

Chinese text images. The detection of curve and its correction thereafter worked very well with all sample images. The experimental results of proposed method are promising as shown in Figures 10 and 12.

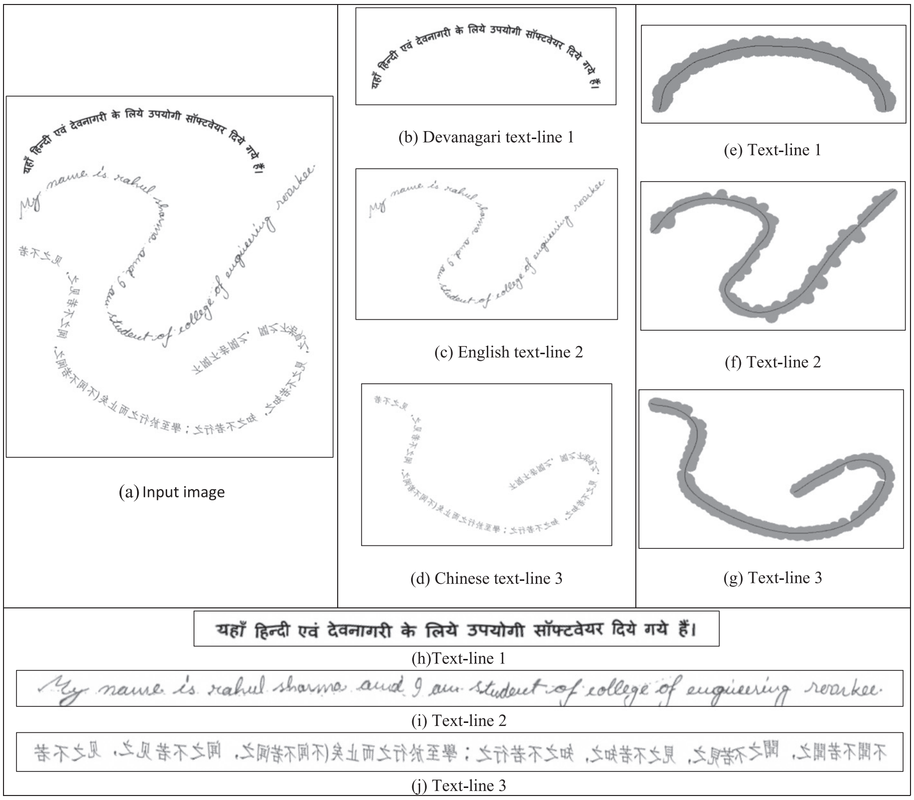
V. Conclusions

In this paper, we presented a curved text-line straightening (correction) technique over the text images having wide variations in terms of font, layout, and size of machine printed as well as handwritten Devanagari, English, and Chinese text-lines. The proposed straightening algorithm is tested on 140 images but more test samples can reveal more output cases in terms of merits or demerits of proposed algorithm. However, the results of few images needs to be further corrected for the better performance of OCR systems. Although we have done experiments only on Devanagari, English, and Chinese text-lines, but proposed algorithm can also be useful to handle all type of languages and scripts such as Brahmi, Grantha, Sinhalese, Bali, etc. The main contribution of this work is in the development of language/script-independent OCR systems.

In contrast, an approach for curved text-line straightening is proposed in this work which can handle multi-font size and type and multi-script text-lines within a single document. The proposed approach is limited to straightening of text-lines and words only and also cannot work well on text-lines not having gap between text-lines. Selection of size of structuring element is set manually in our proposed approach. Automation of structuring element's size selection is required for the enhancement of the accuracy of the proposed approach.

To the best of our knowledge, not a single method proposed to correct the orientation of curved text-line has been reported.

Most of the works reported on Indian languages are on straight text-line documents. Elaborated studies on


यहाँ हिन्दी एवं देवनागरी के लिये उपयोगी सॉफ्टवेयर दिये गये हैं।
My name is rahul sharma and I am student of college of engineering roorkee.

(a) Input image

(b) Devanagari text-line 1

(c) English text-line 2

(d) Chinese text-line 3

(e) Text-line 1

(f) Text-line 2

(g) Text-line 3

(h)Text-line 1

(i) Text-line 2

(j) Text-line 3

Figure 12 The results of multi-script text-line segmentation. (a) Input image. **(b)** Devanagari text-line 1. **(c)** English text-line 2. **(d)** Chinese text-line 3. **(e)** Text-line 1. **(f)** Text-line 1. **(g)** Text-line 3. **(h)** Text-line 1. **(i)** Text-line 1. **(j)** Text-line 3.

curved, multi-oriented, or skewed text-line documents are not much undertaken by the researchers in the development of script/language-independent OCR systems.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Department of Computer Science and Engineering, College of Engineering Roorkee, Roorkee 247667, India. [2]Director (Research), Graphic Era University, Dehradun, Dehradun 248001, India. [3]Department of Electronics and Communication Engineering, IIT Roorkee, Roorkee 247667, India.

Received: 17 October 2013 Accepted: 2 July 2014
Published: 19 July 2014

References

1. BM Singh, A Mittal, D Ghosh, An evaluation of different feature extractors and classifiers for offline handwritten Devanagari character recognition. Journal of Pattern Recognition and Research **06**(2), 269–277 (2011)
2. D Ghosh, T Dube, AP Shivaprasad, Script recognition: a review. IEEE Trans. Pattern Anal. Mach. Intell. **32**, 2142–2161 (2009)
3. U Pal, BB Chaudhuri, Indian script character recognition: a survey. Pattern Recognition **37**, 1887–1899 (2004)
4. D Doermann, J Liang, H Li, Progress in camera-based document image analysis, in *Proceedings of the 7th International Conference on Document Analysis and Recognition (ICDAR), IEEE, Volume 1*, 2003, pp. 606–610
5. N Arica, FT Yarman-Vural, An overview of character recognition focused on off-line handwriting. IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications And Reviews **31**(2), 216–233 (2001)
6. A Amin, Off-line Arabic character-recognition: the state of the art. Pattern Recognition **31**, 517–530 (1998)
7. *ABBYY Software*, www.finereader.com
8. *Free OCR Software*, www.softi.co.uk
9. S Kumano, K Miyamoto, M Tamagawa, H Ikeda, K Kan, Development of container identification mark recognition system. Trans. Inst. Electron. Inform. Commun. Eng. D-I **J84D-II**(6), 1073–1083 (2001)
10. Y Cui, Q Huang, Character extraction of license plates from video, in *Proceeding of International Conference on CVPR*, 1997, pp. 502–507
11. S Hua, W Liu, HJ Zhang, Automatic performance evaluation for video text detection, in *International Conference on Document Analysis and Recognition (ICDAR)* (Seattle, WA, USA, 2001), pp. 545–550
12. AK Jain, B Yu, Automatic text location in images and video frames. Pattern Recognition **31**(12), 2055–2076 (1998)
13. Y Lim, S Choi, S Lee, Text extraction in MPEG compressed video for content-based indexing, in *Proceeding of 15th International Conference on Pattern Recognition (ICPR), Volume 4*, 2000, pp. 409–412
14. T Pavlidis, Recognition of printed text under realistic conditions. Pattern Recognition Letters **14**(4), 317–326 (1993)
15. T Sato, T Kanade, EK Hughes, MA Smith, Video OCR for digital news archives, in *IEEE International Workshop on Content-Based Access of Image and Video Database*, 1998.
16. Q Sun, Y Lu, Text location in camera-captured guide post images, in *Proceeding of Chinese Conference on Pattern Recognition (CCPR), IEEE*, 2010, pp. 1–4
17. C Xilin, Y Jie, Z Jing, A Waibel, Automatic detection and recognition of signs from natural scenes. IEEE Trans. Image Process **13**(1), 87–99 (2004)
18. SM Bascon, SL Arroyo, PG Jimenez, HG Moreno, LF Ferreras, Road-sign detection and recognition based on support vector machines. IEEE Transactions on Intelligent Transportation Systems **8**(2), 264–278 (2007)
19. Q Xie, A Kobayashi, A construction of pattern recognition system invariant of translation, scale-change and rotation transformation of pattern. Transactions of the Society of Instrument and Control Engineers **27**, 1167–1174 (1991)
20. YY Tang, HD Cheng, CY Suen, Translation-ring-projection (TRP) algorithm and its VLSI implementations, in *Character and Hand-writing Recognition, World Scientific*, 1991, pp. 25–56
21. S Adam, JM Ogier, C Carlon, R Mullot, J Labiche, J Gardes, Symbol and character recognition: application to engineering drawing. International Journal of Document Analysis and Recognition **3**, 89–101 (2000)
22. TN Yang, SD Wang, A rotation invariant printed Chinese character recognition system source. Pattern Recognition Letters **22**, 85–95 (2001)
23. H Hase, T Shinokawa, M Yoneda, CY Suen, Recognition of rotated characters by Eigen-space, in *Proceeding of 7th International Conference on Document Analysis and Recognition (ICDAR)*, 2003, pp. 731–735
24. U Pal, N Tripathy, Recognition of Indian multi-oriented and curved text, in *Proceedings of the 8th International Conference on Document Analysis and Recognition (ICDAR)*, 2005, pp. 141–145
25. T Hayashi, N Takagi, A consideration on rotation invariant character recognition, in *World Automation Congress*, 2006, pp. 1–6
26. U Pal, F Kimura, K Roy, T Pal, Recognition of English multi-oriented characters, in *Proceedings of the International Conference on Pattern Recognition*, 2006, pp. 873–876
27. M Monwar, W Haque, PP Paul, A new approach for rotation invariant optical character recognition using Eigen digit, in *Proceedings of the Canadian Conference on Electrical and Computer Engineering*, 2007, pp. 1317–1320
28. PP Roy, U Pal, J Lladós, F Kimura, Convex Hull based approach for multi-oriented character recognition from graphical documents, in *Proceedings of the 19th International Conference on Pattern Recognition (ICPR 2008), IEEE*, 2008, pp. 1–4
29. U Pal, PP Roy, N Tripathy, J Llados, Multi-oriented Bangla and Devnagari text recognition. Pattern Recognition **43**, 4124–4136 (2010)
30. YY Chiang, CA Knoblock, Recognition of multi-oriented, multi-sized, and curved text, in *Proceeding of International Conference on Document Analysis and Recognition (ICDAR)*, 2011, pp. 1399–1403
31. P Shivakumara, TQ Phan, CL Tan, A Laplacian approach to multi-oriented text detection in video. IEEE Trans. Pattern Anal. Mach. Intell. **33**(2), 412–419 (2011)
32. B Gatos, I Pratikakis, SJ Perantonis, Adaptive degraded document image binarization. Pattern Recognition **39**, 317–327 (2006)
33. RC Gonzalez, RE Woods, Digital image processing, in *(DIP/3e), Pearson Education Asia*, 3rd edn., 2008
34. BM Singh, S Goswami, P Goyal, A Mittal, A robust thinning algorithm for straightening of curved text line. In *Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011), December 20–22, Advances in Intelligent and Soft Computing*. Springer **131**, 903–910 (2012)
35. CD Boor, *A practical guide to splines* (Springer Verlag, Berlin - Heidelberg - New York, 1978), pp. 113–114
36. GD Knott, *Interpolating cubic splines* (Springer, Springer, 2000), pp. 151
37. ETY Lee, A simplified B-spline computation routine. Computing Springer-Verlag **29**(4), 365–371 (1982)
38. ETY Lee, Comments on some B-spline algorithms. Computing, Springer-Verlag **36**(3), 229–238 (1986)
39. R Brinks, On the convergence of derivatives of B-splines to derivatives of the Gaussian function. Comp. Appl. Math **27**, 1 (2008)
40. H Prautzsch, W Boehm, M Paluszny, *Bezier and B-spline techniques* (Springer, Berlin-Heidelberg-New York, 2002), pp. 60–66
41. Splitting a uniform B-spline curve, *Splitting a uniform B-spline curve*. (http://www.idav.ucdavis.edu/education/CAGDNotes/Quadratic-Uniform-B-Spline-Curve-Splitting/Quadratic-Uniform-B-Spline-Curve-Splitting.html
42. C Runge, Uber empirische Funktionen und die Interpolation zwischen aquidistanten Ordinaten. Zeitschrift für Mathematik und Physik **46**, 224–243 (1901). available at www.archive.org (http://www.archive.org/details/zeitschriftfrma12runggoog), 1901
43. JP Berrut, LN Trefethen, Barycentric lagrange interpolation. SIAM Review **46**, 501–517 (2004). (http://dx.doi.org/10.1137%2FS0036144502417715), (http://www.worldcat.org/issn/1095-7200.
44. G Dahlquist, A Bjork, Equidistant Interpolation and the Runge Phenomenon, in *Numerical Methods*, 1974, pp. 101–103
45. BS Deyi Zhang, Least squares approximation by splines with free knots: optimization by hybrid of global and local search, in *A Thesis In Mathematics and Statistics*, 2010. 2010

doi:10.1186/1687-5281-2014-36
Cite this article as: Singh *et al.*: **A novel method for straightening curved text-lines in stylistic documents.** *EURASIP Journal on Image and Video Processing* 2014 **2014**:36.