Suman Kumar Bera, Radib Kar, Souvik Saha, Akash Chakrabarty, Sagnik Lahiri, Samir Malakar and Ram Sarkar*

A One-Pass Approach for Slope and Slant Estimation of Tri-Script Handwritten Words

https://doi.org/10.1515/jisys-2018-0105
Received February 21, 2018; previously published online June 27, 2018.

Abstract: Handwritten words can never complement printed words because the former are mostly written in either skewed or slanted form or in both. This very nature of handwriting adds a huge overhead when converting word images into machine-editable format through an optical character recognition system. Therefore, slope and slant corrections are considered as the fundamental pre-processing tasks in handwritten word recognition. For solving this, researchers have followed a two-pass approach where the slope of the word is corrected first and then slant correction is carried out subsequently, thus making the system computationally expensive. To address this issue, we propose a novel one-pass method, based on fitting an oblique ellipse over the word images, to estimate both the slope and slant angles of the same. Furthermore, we have developed three databases considering word images of three popular scripts used in India, namely Bangla, Devanagari, and Roman, along with ground truth information. The experimental results revealed the effectiveness of the proposed method over some state-of-the-art methods used for the aforementioned problem.

Keywords: Slope; slant; handwritten word; oblique ellipse; eigenvector.

1 Introduction

Though the world is rapidly moving towards the electronic era from the traditional manual processing systems, paper documents are still in use for various applications. Digitalization of such documents can bridge the gap between past and present technologies. However, the digitalization of such handwritten documents may become useless if the existing optical character recognition (OCR) system does not convert the corresponding documents into machine-editable form properly. Generally, the task of an OCR system becomes more complex not only for the varying styles of writing but also the skewedness of transcriptions. In handwritten documents, slope and slant are inevitably introduced depending on various factors such as writing style, writing speed, or even the mood of the writer. Such complexities make the OCR process more challenging and result in poor recognition accuracy. Hence, a necessity arises to normalize such documents to an acceptable level so that the recognition system yields optimal outcome.

Slope and slant corrections are the basic pre-processing steps that have been addressed by various researchers in the last few decades. Slope-correction algorithms first measure the angle between the horizontal axis and the line along which the word image is aligned; then, the de-sloped image is generated by rotating the original image inversely according to the slope angle. On the other hand, slant-correction algorithms, in general, first find out the angle between the vertical axis and the most dominant vertical stroke of the slope corrected word, thereafter shearing the image by that angle to provide its de-slanted form.

Various attempts have been taken for page-level skew or text-line level slant correction, where a limited number of works are found for the same at the word level. We present a brief description of recent works related to our problem. Hough transform [8] is one of the most popular techniques used for correcting the

***Corresponding author: Ram Sarkar,** Department of Computer Science and Engineering, Jadavpur University, Kolkata, India, e-mail: raamsarkar@gmail.com
Suman Kumar Bera, Radib Kar, Souvik Saha, Akash Chakrabarty and Sagnik Lahiri: Department of Computer Science and Engineering, Jadavpur University, Kolkata, India
Samir Malakar: Department of Computer Science, Asutosh College, Kolkata, India

slope and slant angles of a word image; however, it becomes computationally expensive when the size of Hough space or data pixels in the word image increases. To cope up with this issue, numerous works [2, 4, 15, 16, 21, 31, 32] have been adopted to reduce the size of Hough space; however, its computational cost remains high. Most recently, Progressive Probabilistic Hough Transform was introduced in Ref. [4] to identify the most prominent lines in a scanned document, and then a special procedure was applied in order to estimate the global skew angle. Limonova et al. [20] used a fast Hough transform to find out the character's slant angle by analyzing its vertical stroke, extracted with the help of the x-derivative of text line in Russian passport recognition. The projection profile-based slope- and slant-correction methods use the projection features of the word image. In their work [5], Cai and Liu computed the slope angle by arbitrarily rotating the document image to fix up the mean square deviation of the projection histogram to be maximized. A major application of projection profile has been found in Refs. [17, 18], where the Winger-Ville distribution was used on both horizontal and vertical projection profiles to find out the slope and slant angles, respectively. However, this approach may suffer from erroneous estimations while handling noisy word images as well as words having condensed regions of ascenders and/or descenders. Haji et al. [12] tried to find the skew angle by splitting the entire word into two vertical slices and then joining their centers. The piece-wise painting algorithm was used in Ref. [1] for skew angle detection. The minimum entropy [10] and maximum variance [28] of the arbitrarily sheared image were used in estimating the slant angles. Papandreou and Gatos [27] centered on core region detection and best-fitting line to find the slant angle. Apart from these, some researchers used the linear regression [11] or maximum eigenvectors of covariance matrix [25] to find a best-fitted line along which the word image is aligned.

From the literature survey, it can be concluded that most of the existing methods are computationally expensive, as they cross over two stages to estimate slope and slant angles. These methods are, in some cases, script dependent [29], which means that a method developed for Matra-based scripts (e.g. Bangla) may not be applicable for non-Matra-based scripts (e.g. Roman) or vice versa. Hence, in a multi-lingual country like India where many scripts are used [23, 24], this kind of script-dependent nature of an algorithm would not serve the practical needs. Furthermore, it may be noted that most of the slant estimation algorithms [18, 27] were developed by conceptualizing words without having any skewedness. However, practically, it is impossible to have such type of de-skewed handwritten text words. To overcome the said problems, two alternatives might be taken: the first one is to rotate the input word image to make its baseline parallel to the X-axis then the slant angle estimation process can be applied on slope-corrected images, and the second one is to find out the slant angle even in the skewed condition. The slant estimation techniques, applied after de-sloping the text word, may suffer from the distortion (salt-and-pepper noise) caused by rotating the word image. In addition to this, it needs two-level de-noising, one after each transformation (i.e. rotation and shear). Along with these, most of the mentioned research works experimented on in-house database and have provided qualitative measures.

As a remedy, we have proposed a one-pass algorithm based on estimating the best-fitted oblique ellipse on the core region of a word image. The selection of the core region not only removes the unbalanced portions [ascendant(s), descendant(s), and elongated character shapes] but also reduces some noise pixels belonging to the portion of the word image beyond the core region (see Figure 1A–C). Moreover, the proposed approach has the potential to work on the handwritten word images written in any script to find out the slope and slant angles most efficiently. Also, we have prepared three databases of isolated handwritten word images written in three different scripts (Bangla, Devanagari, and Roman) and the corresponding ground truth (GT) information (slope and slant angles).

This paper is organized in five sections including this part that narrates the several existing methods related to skew and slant corrections along with a brief introduction to the proposed work. In Section 2, we present the proposed work that describes the estimation and correction processes of word-level obliquity in detail. Section 3 describes the processes of data collection (isolated word samples) and preparation of GT information. The experimental results of the quantitative evaluation process and comparison of our method with some state-of the-art slope and slant estimation methods are presented in Section 4. Finally, the epilogue, followed by future plans, is reported in Section 5.

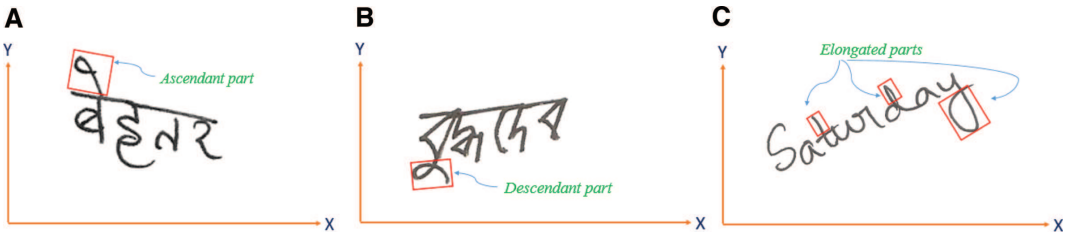

Figure 1: Sample Word Images with (A) Ascendant, (B) Descendant, and (C) Elongated Parts.

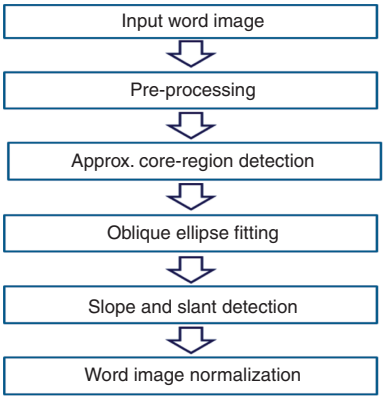

Figure 2: Block Diagram of Proposed Method.

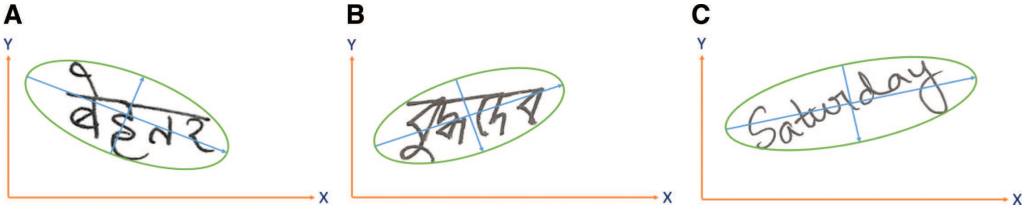

Figure 3: Instances of Erroneous Estimation of Oblique Ellipse Due to (A) Ascendant, (B) Descendant, and (C) Elongated Character Shape.

2 Materials and Methods

This section details the proposed slope- and slant-correction methods. The entire procedure is described in the block diagram shown in Figure 2. Given a handwritten word image, first, we binarize it by using Otsu's thresholding method [26]. Here, the objectives are to estimate the accurate slope (θ) and slant (ϕ) angles in a single attempt, at which a word is originally written. Most of the dictionary words in any language have dimension longer in X-axis than the same in Y-axis, i.e. in general, the ratio of height to width of a word image is <1. Bhowmik et al. [30] suggested that handwritten words, in general, can be best enclosed by an elliptical region to estimate different information about the shape of the word image. Keeping these facts in mind, a best-fit oblique ellipse [3] is conceptualized over the word image for estimating both θ and ϕ. However, the presence of elongated parts of characters in handwritten words may lead to erroneous estimation of such oblique ellipse (see in Figure 3A–C). Therefore, a best-fit straight-line algorithm is applied here prior to ellipse fitting. This method helps in locating the core region of a word image by discarding the other relatively insignificant parts (for slope estimation) of a word image. At the subsequent stage, we use the major and minor axes to find out the longest stroke near the minor axis by using the longest run length of text pixels. Finally, we decide the actual slope and slant angles in reference to the two axes.

Figure 4: Three Different Forms of an Input Word Image.
(A) Original, (B) gray-scale, and (C) binary.

2.1 Pre-processing the Word Images

In the first step, we convert each input 24-bitmap word image $I(x, y)$ in its corresponding gray-scale (say, $W_g(x, y)$) form and then convert it to a binary image (say, $W_m(x, y)$) using Otsu's thresholding method [26]. Next, this binary word image is passed through a morphological close operator [21] with a structuring element of dimensions 3×3 to get rid of unwanted pixels. This image is here termed as $W_b(x, y)$. The gray-scale and binary versions of an input word image are shown in Figure 4B and C.

2.2 Approximate Core Region Detection

Here, the binary word $W_b(x, y)$ is first cropped to its minimal bounding box to avoid extra manipulation required for processing the non-informative part of the word image. Let $\mathcal{B} = \{f(x, y) : (x, y) \in [1, H] \times [1, W]\}$ be the minimally bounded binarized word, where H is the height and W is the width of minimal rectangular bounding box enclosing all the data pixels of $\mathcal{B}$. The values of $f(x, y)$ are "1" and "0," which represent data and non-data pixels, respectively. Using the information of minimal bounding box of $\mathcal{B}$, $W_g(x, y)$ is also cropped into the minimal boundary box to generate a gray-scale word image with minimal boundary (say, G). Therefore, $G = \{g(x, y) : (x, y) \in [1, H] \times [1, W] \wedge g(x, y) \in [0, 255]\}$. The dense region ($D$) of word image is cropped by traversing a structuring rectangle on the minimally bounded binarized word to avoid the non-informative portions of word images.

Also, let $\mathcal{P} = \{(x, y) : (x, y) \in [1, H] \times [1, W] \wedge f(x, y) = 1\}$ be the set of coordinates of data pixels of $\mathcal{B}$. The best-fit straight line ($\mathcal{L}$) of the form $q = mp + c$, where m and c are the slope and the Y intercept value, respectively, is estimated by calculating m and c using the least-square estimating method [30]. We have chosen this algorithm because the best-fitted straight line represents the regression line for a set of random points. Here, consideration of only the data pixel positions of $\mathcal{B}$ as a random variable leads to detecting a $\mathcal{L}$, which partitions the data pixels of it into two regions. The square distances from $\mathcal{L}$ to the data pixels of these two regions become least. The values of m and c are calculated by using the formulas defined below:

$$m = \frac{\sum (x - \bar{X}) \sum (y - \bar{Y})}{\sum (x - \bar{X})^2} \tag{1}$$

and

$$c = \bar{Y} - m\overline{X}, \tag{2}$$

where $\bar{X} = \frac{\sum x}{|\mathcal{P}|}$ and $\bar{Y} = \frac{\sum y}{|\mathcal{P}|}$, and $|\mathcal{P}|$ represents the cardinality of set $\mathcal{P}$.

Distance (d) of every point of $\mathcal{P}$ from line $\mathcal{L}$ is calculated by

$$d = \left| \frac{m\mathcal{P} - q + c}{\sqrt{m^2 + 1}} \right|. \tag{3}$$

Next, the mean (μ_d) and standard deviation (σ_d) of all these distances are determined by

$$\mu_d = \frac{1}{|\mathcal{P}|} \sum d. \tag{4}$$

$$\sigma_d = \sqrt{\frac{1}{|\mathcal{P}|} \sum (d - \mu_d)^2}. \tag{5}$$

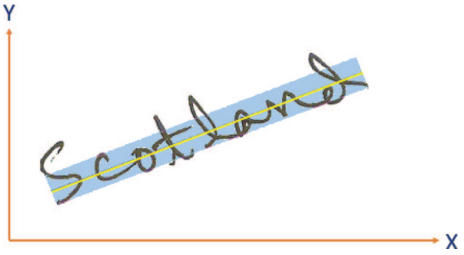

Figure 5: Best-Fitting Line ($\mathcal{L}$) and Approximate Core Region ($\mathcal{P}'$) Marked by Shaded Region.

Now, a new set of points $\mathcal{P}'$, formed from $\mathcal{P}$ that contains all the points, appears within $\mu_d + \rho \times \sigma_d$ distance from $\mathcal{L}$, i.e. $\mathcal{P}' = \{(x', y') : (x', y') \in \mathcal{P} \wedge d \leq \mu_d + \rho \times \sigma_d\}$. Here, ρ is a scalar multiplier. The value of ρ can be adjusted to obtain the core region in a better way. In the current work, the value of ρ is set experimentally. In Figure 5, the approximate core region of $\mathcal{B}$ along with the best-fitted line is shown by the shaded region that contains the pixels belonging to $\mathcal{P}'$. In this figure, $\rho = 0$ is considered. The process of approximate core region estimation is described in Algorithm 1.

Algorithm 1: Approximate core region detection.

Pre-requisite:
$\mathcal{B}_{r*c}$: The binary word image enclosed in a minimum bounding box
Ensure:
C_{p*q}: The core region of $\mathcal{B}_{r*c}$
Procedure: ApproxCoreRegDet($\mathcal{B}_{r*c}$)
1. Identifying the denser region *D* as the concise rectangle, confining dense region of $\mathcal{B}$
   i. $recHeight = avgComponentHeight$, $recWidth = c$, $totalSum = 0$
   ii. **for** $i = 1$ to $(r - recHeight)$ **do**
   iii. Traverse the rectangle from top to bottom and calculate the total pixels Sum_i
   iv. $totalSum = totalSum + Sum_i$
   v. **end for**
   vi. $averageSum = totalSum/(r - recHeight)$
   vii. Get the start and end rows of traversing rectangle for $Sum_i > averageSum$
   viii. Get *D* by cropping $\mathcal{B}$ for the positions start and end rows with width *c*.
2. Approximating core region on *D*
   i. Fit a best line $\mathcal{L}$ on *D* using linear regression
   ii. Calculate Euclidian distance d_n for all *n* pixels from $\mathcal{L}$
   iii. $averageDist = sum(d_n)/n$
   iv. $sdDist = sqrt(sum(pow(d_n - averageDist),2)/n)$
   v. **for** $i = 1$ to *n* **do**
   vi. **if** $d_i > averageDist + rho * sdDist$ **then**
   vii. $C(r, c) = 0$
   vii. **end if**
   ix. **end for**
End Procedure

2.3 Oblique Ellipse Fitting

In this section, an oblique ellipse is fitted to cover most of the data points of $\mathcal{P}'$. The direction of the major axis of this ellipse infers the slope angle. A number of works [3, 9] found in the literature have dealt with fitting an oblique ellipse covering a set of random points. Out of these works, that by Fitzgibbon et al. [9] conceptualized an oblique ellipse from a covariance matrix of random variables. The eigenvector of the largest eigenvalue indicates the direction of the major axis of the fitted ellipse, whereas the direction of minor axis

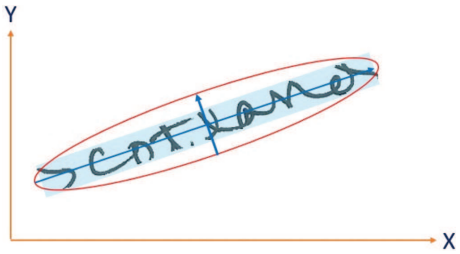

Figure 6: Approximate Core Region ($\mathcal{P}'$) and Best-Fitting Ellipse.

officiates to the eigenvector of the minimum eigenvalue of the covariance matrix. We have chosen this process as it claims that it is a faster technique than the others. In our work, points in $\mathcal{P}'$ are considered as random variables and thereby the covariance matrix (say, C) obtained from points $\mathcal{P}'$ is defined as

$$C(x, y) = E([x - E(x)][y - E(y)]), \tag{6}$$

where $E(x)$ and $E(\mathrm{y})$ are expectations of random variables for x and y coordinates, respectively. Instances of such estimated ellipse on a sample handwritten word are shown in Figure 6. The contour of the oblique ellipse is generated by factorizing C into upper (U)/lower (L) triangular matrix such that $C = U^T U$ or $C = LL^T$ using the Cholesky matrix factorization method, which is faster than the traditional LU factorization method. The process of oblique ellipse fitting [9] is described in Algorithm 2.

Algorithm 2: Oblique ellipse fitting.

Pre-requisite:
x and y:: Two random variables for x and y coordinates
Ensure:
V:: Eigenvector corresponding to a positive eigenvalue
Procedure: ObqElpsFit (x and y)
1. Build design matrix M
   $M = [x. * x\ x. * y\ y. * y\ x\ y\ ones(size(x))]$
2. Build a scatter matrix S
   $S = M' * M$
3. Build a $6 * 6$ constraint matrix N
   $N(6,6) = 0$, $N(1,3) = 2$, $N(2,2) = -1$, $N(3,1) = 2$
4. Solve the eigensystem
   $[gevec, geval] = eig(inv(\mathrm{S}) * N)$
5. Find the positive eigenvalue
   $[PosR, PosC] = find(geval > 0\ \&\ \sim isinf(geval))$
6. Extract the eigenvector corresponding to the positive eigenvalue
   $V = gevec(:, PosC)$
End Procedure

2.4 Finding the Longest Stroke about the Minor Axis

As the word images predictably introduce skewedness, the vertical axis should be corresponded to the minor axis of the ellipse rather the vertical axis of word image itself. Here, the longest stroke near the minor axis is approximated by the run length-based method [13]. Each pixel on the major axis is taken into consideration to find out the longest stroke passing on it; the longest stroke means the longest run of data pixels. For the sake of simplicity and common nature of handwritten text words, we restrict the stroke angle in the range of 45–135° about the major axis, so that any slant angle in between +45° and −45° can be taken care of. The complete procedure for computing the longest stroke and corresponding angle about the minor axis is shown in Algorithm 3.

Algorithm 3: Estimation of the longest stroke angle about the minor axis.

Pre-requisite:
B_{r*c}: The binary word image enclosed in a minimum bounding box
α: The angle between X-axis and the major axis of the fitted ellipse
M_r: The set of row indices corresponding to data pixel positions on the major axis
Ensure:
δ: The angle between the longest stroke and the minor axis of the fitted ellipse
Procedure LongStrkAngEst (B_{r*c}, M_r, α)
1. Skeletonization of B_{r*c} to reduce the computation cost
i. Get the skeletonized image SI_{r*c} corresponding to B_{r*c}
2. Finding the longest run and corresponding angle of individual points on $\mathcal{L}$
i. **for** $i = 1$ to c **do**
ii. $targetPoint = B(M_r, i)$, $tempCount = 0$, $tempTh = 0$
iii. **if** $\alpha \geq 90$ **then**
iv. **for** $th = \alpha + 45$ *to* $\alpha + 135$ **do**
v. Detect all points of line passing through *targetPoint* and $angle = th$
vi. Get the maximum continuous text pixels, *runLength* on the line
vii. **if** $tempCount < runLength$ **then**
viii. $tempCount = runLength$, $tempTh = th$
ix. **end if**
x. **end for**
xi. **else**
xii. **for** $th = \alpha + 45$ to $\alpha + 13$ **do**
xiii. Detect all points of line passing through *targetPoint* and $angle = th$
xiv. Get the maximum continuous text pixels *runLength* on the line
xv. **if** $tempCount < runLength$ **then**
xvi. $tempCount = runLength$, $tempTh = th$.
xvii. **end if**
xviii. **end for**
xix. **end if**
xx. $record(1, i) = th$
xxi. $record(2, i) = tempLength$
xxii. **end for**
xxiii. **for** $i = 3$ to $c - 3$ **do**
xxiv. $record(3, i) = record(2, i-2) + record(2, i-1) + record(2, i) + record(2, i+1) + record(2, i+2)$
xxv. **end for**
3. Finding the longest run length and the corresponding angle
i. $temp = 0$;
ii. **for** $i = 1$ to c **do**
iii. **if** $temp < record(3, i)$
iv. $temp = record(3, i)$
v. $angle = record(1, i)$
vi. **end if**
vii. **end for**
End Procedure

2.5 Estimation of Slope and Slant Angles

The directions of major and minor axes of the ellipse are used to estimate the slope angle (θ) and slant angle (ϕ), respectively, as shown in Figure 7A and B. Let α be the angle (in degree) formed by the major axis and β be the angle (in degree) formed by minor axis with the positive direction of X-axis. α and β are estimated from eigenvectors corresponding to the largest and smallest eigenvalues of C. The slope angle (θ) is the angle formed by the major axis with the positive direction of X-axis. Therefore, θ can be calculated as

$$\theta = \begin{cases} +\alpha, & \text{if } 0 \leq \alpha \leq 90 \\ -(180 - \alpha), & \text{if } 90 < \alpha \leq 180 \end{cases} \tag{7}$$

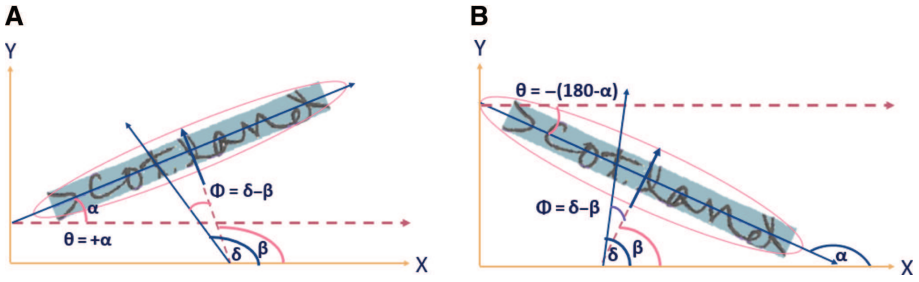

Figure 7: Calculation of Slope Angle (θ) and Slant Angle (ϕ).

Figure 8: Sample Outputs of the Proposed Approach.
(A) Input word, (B) slope-corrected word, and (C) slant-corrected word.

ϕ is the angle between the minor axis and longest stroke near it. Let this stroke form an angle (δ) with the positive direction of X-axis (see Figure 7). Therefore, ϕ is calculated as

$$\phi = \delta - \beta, \tag{8}$$

$$\text{where } \beta = \begin{cases} 90 + \theta, & \text{if } 0 < \alpha \leq 90 \\ \beta = \theta - 90, & \text{otherwise} \end{cases}. \tag{9}$$

2.6 Correction of Slope and Slant

At this stage, first, G is rotated at angle $-\theta$ using affine transformation. New pixel positions after affine transformation are decided by applying the cubic interpolation algorithm [19]. Next, the slope-corrected word is passed through the slant-correction mechanism. Slant correction is carried out by sharing the image at angle $-\phi$. Figure 8 shows the slope-corrected (B) and subsequently the slant-corrected (C) word images.

3 Database and GT Preparation

Database plays a vital role in any document image processing research. The methods proposed in the literature for word-level slope and slant correction are mostly experimented on in-house databases. Few authors have also claimed about preparing GT information to assess their proposed mechanism quantitatively. However, none of such databases as well as GT is freely available for further experiments. Therefore, we have prepared three databases consisting of sloped and slanted handwritten word images of three different scripts.

3.1 Database Preparation

To test the script-invariant nature of the proposed algorithm, we have collected word images of the three most popular scripts used in India, namely Bangla, Devanagari, and Roman. The first two scripts are "Matra"-based scripts while the last one is a "non-Matra"-based script. Words in the database have been written by several writers belonging to different age groups, starting from school-going children to elderly people. Different

Figure 9: Sample Word Images Taken from Our Three Benchmark Databases.

people have different writing styles, and all these variations make the database realistic as well as make it challenging to prove the robustness of our algorithm.

To collect the writing samples, we made 30 equal-sized blocks in A4-sized white sheet, and the writers were requested to write inside the blocks. They had some restrictions in choosing the words to write; for example, non-dictionary words were avoided. Once these filled-in A4 sheets were scanned in 300 dpi as 24bit RGB images (.bmp), individual words were cropped programmatically from the scanned sheets. We have considered 250 handwritten words for each of the said scripts, which means the three databases contain a total of 750 isolated handwritten word samples. The databases are named as Database-A, -B, and -C for Bangla, Devanagari, and Roman scripts, respectively. Some of the sample images taken from the databases are shown in Figure 9. It has been noticed that most of the word samples are skewed as well as slanted naturally.

3.2 GT Preparation

To the best of our knowledge, proper slope and slant estimations of freestyle words can only be done by human beings. Thus, we have prepared the GT (i.e. slope and slant angles) of the handwritten words in a semi-automatic way. To determine the slope angle, we first used the algorithm defined by Gupta and Chanda [11], wherein we found proper results for most of the cases, measured in terms of visual perception. The remaining cases where the resultant angles were not as predicted were handled using a well-known tool, IrfanView444. Similarly, to determine the slant angle, we first used the algorithm defined by Kavallieratou et al. [18]; then, the erroneous cases were dealt with by manually shearing the image by different angles. In most cases, it has been seen that the skew angles were varied from 0° to 25°, whereas the slant angles were varied from 0° to 40° on analysis of 750 handwritten words.

4 Experimental Results

We have implemented our algorithm using MATLAB2013 software on a PC with 6 GB RAM and Intel Core i3-2328M CPU @ 2.20 GHz processor. For the assessment of a slope/slant angle estimation algorithm, we have relied only on quantitative analysis. The evaluation parameter considered here is the absolute error from the GT angle. On the other hand, for the assessment of the correction algorithms of the said problems, we have relied on the qualitative outcome of the same.

4.1 Evaluation Process

The resultant slope and slant angles from an estimation technique are compared with the corresponding GT angles. Let the slope angle estimated by some slope detection algorithm and GT slope angle information for a word image be θ_R and θ_{GT}, respectively. Therefore, error in detection by this technique for the word image under consideration is calculated as $AE = |\theta_R - \theta_{GT}|$, where the function $|x|$ returns the absolute value of the variable x and AE indicates the absolute error. Finally, the estimation parameter, which is considered here as average error (μ_{AE}), is calculated as

$$\mu_{AE} = \frac{1}{n}\sum_{i=1}^{n} AE_i, \tag{10}$$

where n and AE_i are the total number of words considered here and absolute error for the i^{th} word in the database. Thus, a smaller average error implies more accuracy. A similar calculation is done for the assessment of slant angle detection method.

4.2 Parameter Selection for Core Region Estimation

As mentioned in Section 2.2, a constant multiplier ρ is used to obtain an approximate core region. It is worth mentioning that better approximation of the core region would lead to better estimation of the slope and slant angles. Therefore, to set the optimal value of ρ, we have randomly selected 100 sample words from each of the above-mentioned databases. Next, we applied the present slope angle detection mechanism with varying ρ values. The experimental outcome is depicted in Table 1. The average errors in detecting slope angle (μ_{AE}^{Slope}) are recorded for ρ values of 1.5, 1.5, and 1.0 for the databases A, B, and C, respectively. These values were used for the rest of the experiments.

4.3 Performance of the Proposed Method on Noisy Data

We performed a set of experiments to check how the present technique performs under a noisy environment. We added three different kinds of noises, namely salt and pepper, Gaussian, and Poisson, to gray-scale word images. Instances of a gray-scale image with added noise are shown in Figure 10B–D. Then, these noisy word images were processed using the mechanism described in Section 2.1 to obtain the binary form. Instances of such binary versions of the noisy word images are depicted in Figure 10. The experimental outcomes are shown in Table 2, which confirms that the proposed method performs well even under a noisy environment.

Table 1: Average Error Recorded for Varying Values of ρ.

ρ Values	Average absolute error (μ_{AE}^{Slope})		
	Database-A	**Database-B**	**Database-C**
2.0	NA	NA	NA
1.5	**2.385**	**5.227**	4.546
1.0	2.541	5.481	**4.444**
0.5	2.864	6.094	4.717
0.0	3.150	6.141	5.189
−0.5	3.212	5.467	5.602
−1.0	NA	5.884	NA
−1.5	NA	NA	NA

Bold style numbers indicate best scores.

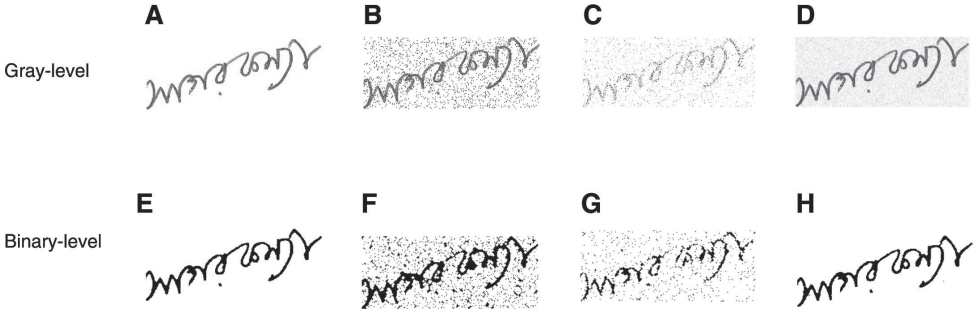

Figure 10: Instances of Gray-Level and Preprocessed Word Samples after Adding (A, E) No Noise (i.e. Actual), (B, F) Salt and Pepper, (C, G) Gaussian, and (D, H) Poisson.

Table 2: Performances on Noisy Word Samples.

Databases	Actual	Noise added				
		Salt and pepper with density		Gaussian with Mean, SD		Poisson
		5%	10%	0, 0.01	0.5, 0.1	
	Average slope angle detection error					
Database-A	2.916	3.207	3.851	3.092	3.193	2.981
Database-B	3.904	4.122	4.313	4.055	3.979	3.954
Database-C	4.017	4.447	5.261	4.053	4.436	4.026
	Average slant angle detection error					
Database-A	3.078	3.179	3.211	4.102	3.445	3.108
Database-B	2.758	2.889	3.012	4.227	4.012	2.808
Database-C	3.018	3.109	3.217	4.099	3.655	3.133

Table 3: Comparison of the Proposed Method with State-of-the-Art Methods.

	Methods	Database-A	Database-B	Database-C
Slope angle detection	Gupta and Chanda [11]	4.364	4.876	4.916
	Malakar et al. [21]	4.311	4.093	7.456
	Okun et al. [25]	4.315	4.898	4.738
	Proposed	**2.916**	**3.904**	**4.017**
Slant angle detection	Pastor et al. [28]	3.276	3.228	4.216
	Kavallieratou et al. [18]	3.622	3.011	3.254
	Papandreou and Gatos [27]	**2.977**	2.957	3.102
	Proposed	3.078	**2.758**	**3.018**

Bold style numbers indicate best scores.

4.4 Comparison with State-of-the-Art Methods

Most of our collected word samples are sloped as well as slanted in nature. The present algorithm takes all such original word images as inputs and subsequently corrects the slope and slant angles. We implemented three slope-correction and three slant-correction techniques [11, 18, 21, 25, 27, 28] to make a comparison with state-of-the-art slope/slant angle estimation methods. For all these algorithms, we calculated the average error for both cases in degrees. Table 3 demonstrates that our method outperforms the state-of-the-art slope- and slant-correction methods considered.

The pictorial disparity of state-of-the-art methods with the proposed method, shown in Figure 11, indicates that the outcomes of the proposed method are much closer to the GT word images than the others.

4.5 Performance on Digit String

The robustness and generalness of the present slope- and slant-correction technique can be established if the same is performed equally well on digit string or numeral word images [22]. This is why we conducted an experiment on handwritten digit string samples taken from an openly available dataset, which is made available to the research community through the Handwritten Document Recognition Competition (HDRC) 2013 [7]. HDRC 2013 was organized in conjunction with the International Conference on Document Analysis and Recognition 2013 for the recognition of handwritten digits. This dataset contains 1262 numeral strings. Of these, we have selected 100 words that are mostly slanted in nature. Next, each of these digit strings were rotated at different angles and then fed as input to our method. The outcomes are really impressive. Some sample outputs are shown in Figure 12.

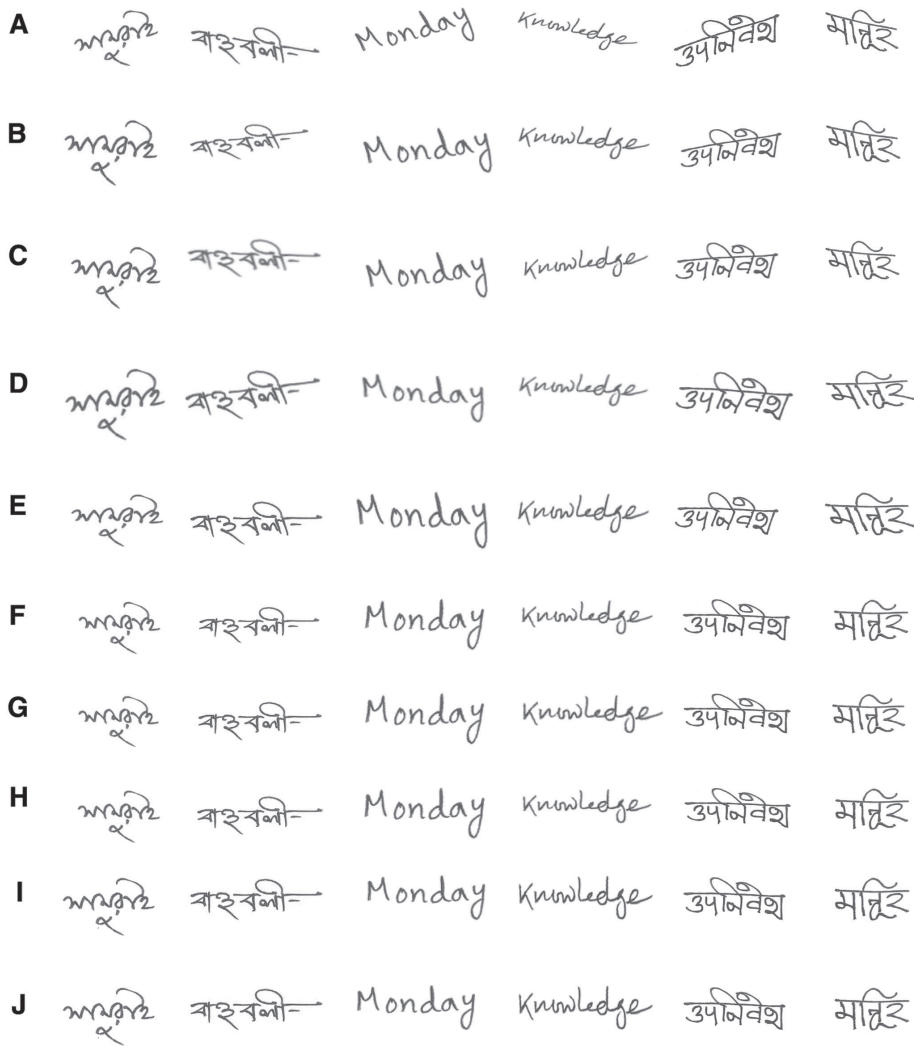

Figure 11: Each Row of the Figure Represents (A) Sample Words; Slope-Corrected Words by (B) Gupta and Chanda [11], (C) Malakar et al. [21], and (D) Okun et al. [25]; (E) Proposed Slope-Correction Approach; Slant-Corrected Words by (F) Pastor et al. [28], (G) Kavallieratou et al. [18], and (H) Papandreou and Gatos [27]; (I) Proposed Slant-Correction Approach; and (J) GT words.

Figure 12: Each Row of the Figure Represents the Performance of our Proposed Method on Digit Strings. (A) Three sample images. (B) Outputs after skew correction. (C) Final outputs after slant correction.

4.6 Error Analysis

Although our approach produced impressive results in almost every case, it failed in two exceptional cases: (i) when the height of the core region of a given word exceeds its width, the eigenvector of the largest eigenvalue in the covariance matrix cannot fit the oblique ellipse properly over the word image. The same may

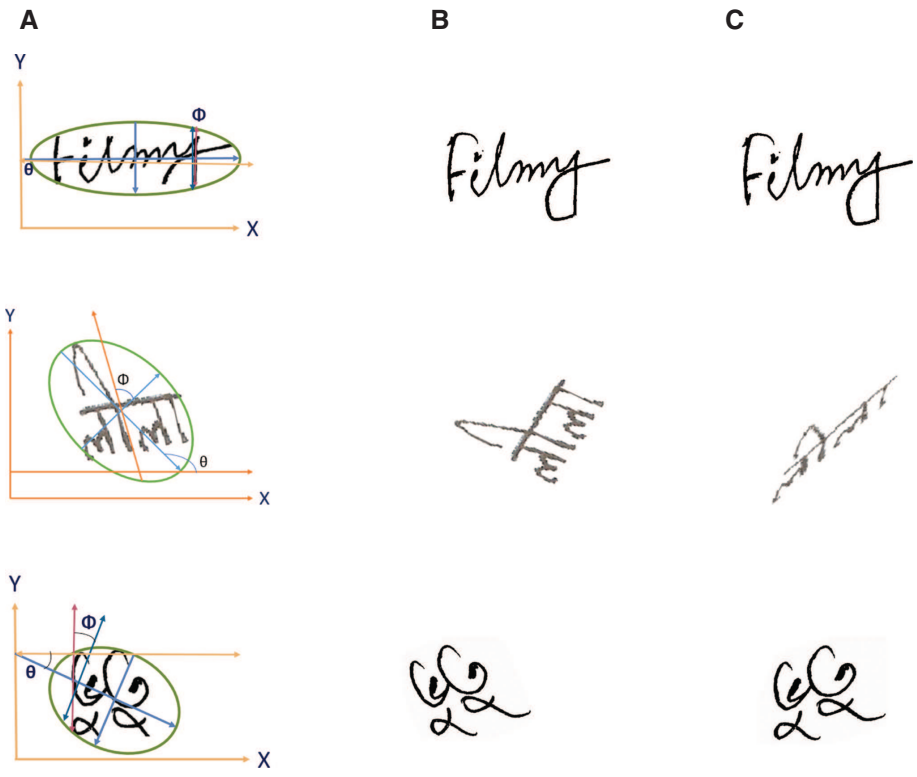

Figure 13: Limitations of Our Approach.
(A) Improper ellipse fitting on approximate core region of input word. (B) Results after slope correction. (C) Results after slant correction of two critical cases.

occur in erroneous estimation of the core region when the horizontal pixel density looks the same altogether, and (ii) the absence of strong near vertical strokes makes an improper estimation of the slant angle. Figure 13 shows some examples where a pictorial explanation of these shortcomings of the proposed method can be understood easily.

It is to be noted that our method was designed to be useful for tri-script text words, and it deals with almost all possible complexities observed therein. The above argument on error analysis reveals that some of the bottlenecks found in this method occur due to the improper formation of ellipse and the absence of prominent near-vertical strokes with respect to the minor axis of the fitted ellipse. Due to the freestyle writings, these cases may occur in any text word irrespective of the script in which it is written.

5 Conclusion

In this paper, we have presented a simple but effective method for slope and slant estimations of handwritten word images. The uniqueness of the approach is that it is a one-pass approach used to handle slope as well as slant of both Matra-based and non-Matra-based word images. The comparative study with some existing methods reveals that the proposed method is rather efficient as well as effective in finding out the slope and slant angles of unconstrained handwritten text words. Hence, it can be said that this algorithm would certainly make the subsequent OCR or word recognition much easier. In the future, we must address the remaining flaws in our algorithm, like a concise ellipse fitting in accurate direction and best ever core region detection, even in noisy environments. An adequate binarization like that in Refs. [6, 14] will then be needed for binarization of such noisy word images. We also plan to incorporate more sample words for different scripts in the database to establish the robustness of the proposed method.

Bibliography

[1] A. Alaei, U. Pal, P. Nagabhushan and F. Kimura, A painting based technique for skew estimation of scanned documents, in: *Proceedings of the 2011 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 299–303, IEEE, 2011.

[2] A. Amin and S. Fischer, A document skew detection method using the Hough transform, *Pattern Anal. Appl.* **3** (2000), 243–253.

[3] S. Bhowmik, S. Malakar, R. Sarkar and M. Nasipuri, Handwritten Bangla word recognition using elliptical features, in: *International Conference on Computational Intelligence and Communication Networks (CICN)*, November 2014, pp. 257–261, IEEE, 2014.

[4] O. Boudraa, W. K. Hidouci and D. Michelucci, An improved skew angle detection and correction technique for historical scanned documents using morphological skeleton and progressive probabilistic Hough transform, in: *5th International Conference on Electrical Engineering-Boumerdes (ICEE-B)*, October 2017, pp. 1–6, IEEE, 2017.

[5] J. Cai and Z. Q. Liu, Off-line unconstrained handwritten word recognition, *Int. J. Pattern Recogn. Artif. Intell.* **14** (2000), 259–280.

[6] B. Das, S. Bhowmik, A. Saha and R. Sarkar, An adaptive foreground-background separation method for effective binarization of document images, in: *International Conference on Soft Computing and Pattern Recognition*, December 2016, pp. 515–524, Springer, Cham, 2016.

[7] M. Diem, S. Fiel, A. Garz, M. Keglevic, F. Kleber and R. Sablatnig, ICDAR 2013 Competition on Handwritten Digit Recognition (HDRC 2013), in: *12th International Conference on Document Analysis and Recognition (ICDAR)*, August 2013, pp. 1422–1427, IEEE, 2013.

[8] R. O. Duda and P. E. Hart, Use of the Hough transformation to detect lines and curves in pictures, *Commun. ACM* **15** (1972), 11–15.

[9] A. W. Fitzgibbon, M. Pilu and R. B. Fisher, Direct least square fitting of ellipses, *IEEE Trans. Pattern Anal. Mach. Intell.* **21** (1999), 476–480.

[10] B. Gatos, I. Pratikakis, A. L. Kesidis and S. J. Perantonis, Efficient off-line cursive handwriting word recognition, in: *Tenth International Workshop on Frontiers in Handwriting Recognition*, October 2006, Suvisoft, 2006.

[11] J. D. Gupta and B. Chanda, An efficient slope and slant correction technique for off-line handwritten text word, in: *Fourth International Conference of Emerging Applications of Information Technology (EAIT)*, December 2014, pp. 204–208, IEEE, 2014.

[12] S. A. B. Haji, A. James and S. Chandran, A novel segmentation and skew correction approach for handwritten Malayalam documents, *Proc. Technol.* **24** (2016), 1341–1348.

[13] S. C. Hinds, J. L. Fisher and D. P. D'Amato, A document skew detection method using run-length encoding and the Hough transform, in: *Proceedings of the 10th International Conference on Pattern Recognition*, 1990.

[14] P. Jana, S. Ghosh, S K. Bera and S. R. Sarkar, Handwritten document image binarization: an adaptive K-means based approach, in: *IEEE Calcutta Conference (CALCON)*, December 2017, pp. 226–230. IEEE, 2017.

[15] H. F. Jiang, C. C. Han and K. C. Fan, A fast approach to the detection and correction of skew documents, *Pattern Recogn. Lett.* **18** (1997), 675–686.

[16] T. Jipeng, G. Hemantha Kumar and H. K. Chethan, Skew correction for Chinese character using Hough transform, *Int. J. Adv. Comput. Sci. Appl. – IJACSA (Special Issue)* (2011), 45–48.

[17] E. Kavallieratou, N. Fakotakis and G. Kokkinakis, Skew angle estimation for printed and handwritten documents using the Wigner-Ville distribution, *Image Vis. Comput.* **20** (2002), 813–824.

[18] E. Kavallieratou, N. Fakotakis and G. Kokkinakis, Slant estimation algorithm for OCR systems, *Pattern Recogn.* **34** (2001), 2515–2522.

[19] R. Keys, Cubic convolution interpolation for digital image processing, *IEEE Trans. Acoust. Speech Signal Process.* **29** (1981), 1153–1160.

[20] E. Limonova, P. Bezmaternykh, D. Nikolaev and V. Arlazarov, Slant rectification in Russian passport OCR system using fast Hough transform, in: *Ninth International Conference on Machine Vision (ICMV 2016)*, March 2017, vol. 10341, p. 103410P, International Society for Optics and Photonics, 2017.

[21] S. Malakar, B. Seraogi, R. Sarkar, N. Das, S. Basu and M. Nasipuri, Two-stage skew correction of handwritten Bangla document images, in: *Third International Conference on Emerging Applications of Information Technology (EAIT)*, November 2012, pp. 303–306, IEEE, 2012.

[22] S. M. Obaidullah, C. Halder, N. Das and K. Roy, A new dataset of word-level offline handwritten numeral images from four official Indic scripts and its benchmarking using image transform fusion, *Int. J. Intell. Eng. Inform.* **4** (2016), 1–20.

[23] S. M. Obaidullah, K. C. Santosh, C. Halder, N. Das and K. Roy, Automatic Indic script identification from handwritten documents: page, block, line and word-level approach, *Int. J. Mach. Learn. Cybern.* in press. (2017), 1–20.

[24] S. M. Obaidullah, C. Halder, K. C. Santosh, N. Das and K. Roy, PHDIndic_11: page-level handwritten document image dataset of 11 official Indic scripts for script identification, *Multimed. Tools Appl.* **77** (2018), 1643–1678.

[25] O. Okun, M. Pietikäinen and J. Sauvola, Document skew estimation without angle range restriction, *Int. J. Doc. Anal. Recogn.* **2** (1999), 132–144.
[26] N. Otsu, A threshold selection method from gray-level histograms, *IEEE Trans. Syst. Man Cybernet.* **9** (1979), 62–66.
[27] A. Papandreou and B. Gatos, Slant estimation and core-region detection for handwritten Latin words, *Pattern Recogn. Lett.* **35** (2014), 16–22.
[28] M. Pastor, A. Toselli and E. Vidal, Projection profile based algorithm for slant removal, in: *International Conference Image Analysis and Recognition*, September 2004, pp. 183–190, Springer, Berlin, Heidelberg, 2004.
[29] R. Sarkar, S. Malakar, N. Das, S. Basu and M. Nasipuri, A script independent technique for extraction of characters from handwritten word images, *Int. J. Comput. Appl.* **1** (2010), 85–90.
[30] A. Savitzky and M. J. Golay, Smoothing and differentiation of data by simplified least squares procedures, *Anal. Chem.* **36** (1964), 1627–1639.
[31] C. Singh, N. Bhatia and A. Kaur, Hough transform based fast skew detection and accurate skew correction methods, *Pattern Recogn.* **41** (2008), 3528–3546.
[32] S. N. Srihari and V. Govindaraju, Analysis of textual images using the Hough transform, *Mach. Vis. Appl.* **2** (1989), 141–153.