International Journal of Innovative
Computing, Information and Control
Volume **8**, Number **5(A)**, May **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 3149–3161

FUZZY BASED PREPROCESSING USING FUSION OF ONLINE AND OFFLINE TRAIT FOR ONLINE URDU SCRIPT BASED LANGUAGES CHARACTER RECOGNITION

Muhammad Imran Razzak[1], Syed Afaq Husain[2]
Abdulrahman A. Mirza[3] and Abdel Belaïd[4]

[1]Department of Computer Science and Engineering
Air University
Islamabad, Pakistan
imran.mian@yahoo.com

[2]Riphah International University
Rawalpindi, Pakistan
drafaqh@gmail.com

[3]Information System Department
King Saud University
P.O.Box 51178, Riyadh 11543, Saudi Arabia
amirza@ksu.edu.sa

[4]READ, LORIA, France

Received August 2010; revised July 2011

Abstract. *Character recognition has been fascinating and intense field of pattern recognition research since early days of computer. This task becomes more challenging when it involves Urdu script based languages especially written in handwritten Nasta'liq font due to the large variations and complexity of the script. Fuzzy logic is an important tool to deal with vague, incomplete, noisy and contradictory information. In order to make handwritten communication with machine more natural, we propose several preprocessing steps for handwritten online Urdu script character recognition to overcome the issue in raw input. We present fuzzy based several preprocessing operations in order to normalize the handwritten stroke using both online and offline domain. The proposed technique is also the necessary step towards character recognition, person identification, personality determination where input data are processed from all perspectives.*
Keywords: Preprocessing, Baseline, Slant correction, Character recognition, Smoothing, Urdu, Nasta'liq, Naskh

1. **Introduction.** Character recognition has been an ongoing field of research since the early days of computer and remains a challenging issue in the field of pattern recognition due to complexities involved in it especially in handwritten character recognition. With respect to mode of input, character recognition is classified into two classes: offline and online. In offline, the input is spatial coordinates in the form of image whereas in online, sequence of points with embedded pen up and down. This additional information makes online character recognition a little easier than offline. Handwritten provides the most promising method for interaction with small portable machine. From the last few decades, online character recognition is getting more popularity due to increasing popularity of hand-held devices and natural way of input to the machines [26].

Urdu scripts are followed by more than 1/4th population of the world in the form of many languages, i.e., Arabic, Persian, Urdu, Punjabi, Pashto etc. in many countries [17]. Urdu consists of 58 alphabets and the ghost shapes of Urdu alphabets also present in other

Arabic script based languages shown in Figure 1. Moreover, Urdu is written in Nasta'liq style which is more complex than any other style followed by other Arabic script based language; i.e., Nasta'liq contains 32 shapes of "ت" depending upon the attached character on both side whereas Naskh consists of only four shapes. Thus, Urdu is more complicated in its family due to Nasta'liq writing style shown in Figure 5. Urdu script based languages are written in cursive style from right to left and are very rich in diacritical marks. It is also the context sensitive language and written in the form of ligatures which comprise a single or many different characters. Most characters have different shapes depending on their position and their adjoining character in the word and characters overlap each other, while studies dealing with Arabic script based languages characters are very less especially for Urdu script.

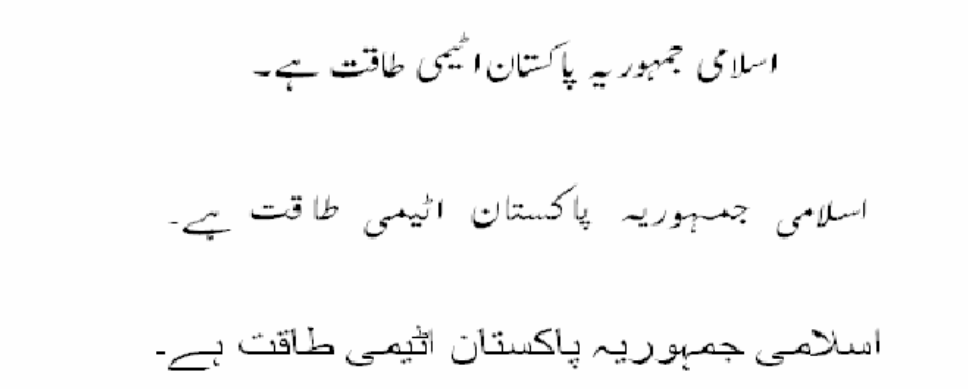

FIGURE 1. Urdu samples in three different styles: Urdu Nasta'liq, Urdu Naskh

The accuracy of text recognition either by human or machine is highly dependent on the quality of input. Preprocessing is one of the important phases of Urdu character recognition and has direct influence on recognition result. It is a compulsory part in order to compensate the variations in handwritten stroke. We present several preprocessing steps by combining both offline and online. Proposed preprocessing from two domains in turn provides compensation for variability in size, smoothing from irregular pattern, skew detection and base line estimation, etc.

2. **Related Work.** On-line Latin or Asian language has been a research issue since thirty years [25]. However, only few researches focused on Arabic language. Most of the Arabic script recognition systems do not allow noisy input. Therefore, multi-cultural and multi-language handwriting styles are the current and hot research issue for Arabic script based languages. In order to simplify the input text to improve the recognition accuracy, several processing techniques for both online and offline character recognition have been proposed. Basically the basic aim of the preprocessing steps is to reduce the variations and prepare the character shape for segmentation, feature extraction, etc. Unlike English, Japanese, etc., the cursive Arabic script has not received much attention during the last years [1]. Furthermore, within the family of Arabic script online handwritten character recognition, studies dealing with Urdu Nasta'liq script is scarce.

The main objective of the preprocessing steps is to normalize the handwritten strokes and to reduce variations that would otherwise complicate the recognition and reduce its rate. Smoothing and de-hooking is performed before feature extraction as a preprocessing step [2,3]. As the input text contains irregularity due to the natural way of writing, the preprocessing techniques perform 2 to 3 pixel smoothing and de-hooking on the Urdu input text. S. Malik and S. A. Khan [4] performed repetition removal and filtering step

in preprocessing phase whereas M. Hussain [5] calculates the only displacement from 4,8,16 displacements between two points. K. Al-Ghoneim et al. [6] performed translation, scaling, connected line generation and smoothing. They compute the image's center of gravity and translate the image such that its origin is the center of gravity. They scale the image so that the maximum radius for the character pixels equal to half the grid size. The radius of a pixel is defined as the length of the straight line connecting the pixel to the origin and connected line generation: they used the Bresenham's line algorithms to fill the missing points that was left due to fast pen movement and finally, smooth the input curve by inspecting each subsequence of pixels and replacing it by a shorter version. F. Biadsy et al. worked on geometrical processing phase to minimize the handwriting variations [7]. They used low pass filter algorithm to reduce the noise and

to remove the imperfections caused by acquisition device. To eliminate the redundant points that are irrelevant for classification, Douglas and Peucker's algorithm [8] was used. The document is broken into text lines and words. The handwriting text line extraction techniques for on-line, depending on the y-axis histogram projection and character geometry (width, height, etc.), does not function well on Arabic handwriting due to its characteristics. This technique is more suitable to the Arabic language nature [9]. F. Bouchareb et al. presented different preprocessing techniques for handwritten Arabic words using Hough transformation and geometrical processing [1]. The authors performed skeletonization, smoothing, slant correction and baseline estimation. Hough transformation is used for the baseline estimation while for slant estimation, rules are used. The slant is estimated by analyzing the top most endpoints of the baseline and replaces the points that belong to the same considered endpoints. H. Al-Rashaideh presented several preprocessing steps for online Arabic handwritten recognition [11].

Baseline is the virtual line on which semi cursive or cursive text are aligned and/joined. Generally baseline is kept in mind during both writing and reading. Baseline detection is not only used for automatic character recognition but it is also necessary for human reading. Without baseline detection it is very difficult to read the text even for human and error rate increase up to 10% while the context sensitive interpretation is involved. Whereas in automatic classification no context based interpretation is involved, thus baseline detection is the necessary part of better classification especially for Arabic script based languages. The detection of diacritical marks is not easy task without baseline. Several baseline detection methods based on horizontal projection have been proposed in the literature but they are for large text lines.

Boubaker et al. presented a novel method for both online and offline Arabic handwritten text [12]. S. S. Maddouri and H. E. Abed compared the six methods for Arabic character recognition on IFN/ENIT database [13]. Projection based method fails to estimate the baseline for short word length and words having more diacritical marks, ascender and descender. Min-Max and PAWs are presented based on the projection based method. Min-Max contour method using diacritic points from the word contour and two baselines upper and lower are extracted from the mean of maxima and minima respectively. The combination of Min-Max and some structural primitives, i.e., loops, diacritical marks is used for baseline estimation. These additional primitives are used to differentiate the contours. Faisal et al. modified the RAST algorithms by introducing two descender lines d1 and d2 for Urdu images [14]. Farooq et al. presented a linear regression on local minima of word contour for the baseline detection [15]. Horio et al. used the relative position of a character and perform several preprocessing steps in order to normalize the handwritten strokes [16]. R. A. Mohammad et al. presented a vertical projection algorithm obtained by summing the value along x-axis and detected two baselines. The lower baseline is identified by the analysis of the maximum projection profile. The upper

baseline is estimated by scanning the image from top to bottom [18]. M. Razzak et al. extracted the baseline by computing the minimum enclosing rectangle and drawing vertical projection [17]. Alkhateeb presented knowledge based baseline estimation by using the location information for the baseline estimation. The algorithm is improved by estimating the baseline at the bottom half of the image because of baseline existence at bottom of word [19].

Gabor filter has been used for many applications of image processing, i.e., edge detection, writer identification, texture processing, etc. [20]. The vertically projection is inefficient for small length text. Benouareth et al. used the projection after transforming the image into Hough parameters for the baseline estimation [21]. M. Pechwitz and V. Maergner used linear piecewise curves using projections for baseline estimation [22]. M. Razzak et al. performed fuzzy based biologically technique which inspired several preprocessing steps [27]. The layered based preprocessing steps are performed locally and some additional knowledge from the previous word is used as similar to human visual system.

The study presented [2-5,7-10], performed preprocessing only on the online side, are smoothing ,de-hooking [2,3] and connected line generation [6] while Somaya [10] worked on stroke normalization. The existing techniques are on online information while they did not focus the offline processing on online data, which are base line finding, combining strokes and skews correction. Due to the complexity of Urdu text, it is better to utilize the offline processing along with online processing for online input text.

3. **Proposed Preprocessing.** The following are the preprocessing steps applied to normalize the Urdu handwritten stroke.

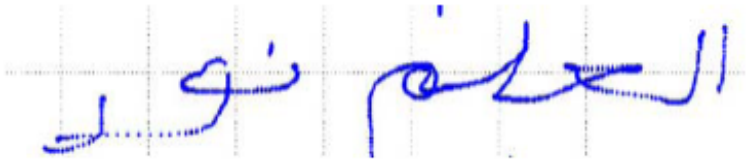

FIGURE 2. Noise during pen lifting and pen down [24]

3.1. **De-hooking.** Hooks are most common artifacts that occur at the beginning and the ending during the handwriting, commonly generated durin‘g’ the pen-down and pen-up movements shown in Figure 2. The presence of hooks may lead to incorrect feature extraction and create problem during character recognition for characters that start with jeem ‘ج’ and ayen ‘ع’ due to naturally presence of hooks. Thus, before removing the hooks, it is necessary to take the decision whether it belongs to hook or not. It may possible that small up of the stroke ‘ج’ is removed instead of hook which is the most important part in the detection of stroke ‘ج’ as shown in Figure 3. To avoid de-hooking in jeem, de-hooking at beginning is not performed on those ligatures which are written from left to right for some length like ‘ج’. The isolated jeem is written from left to right and the ligature ‘جر’ is also written from left to right at the beginning while the remaining ligature is written from right to left. For other strokes, if the variation in the chain code of length 6 at the beginning or end is less than the specified threshold (T = 4), then that part, considered as hook, is removed by either discarding it or replacing the respective co-ordinates with the neighbor ones.

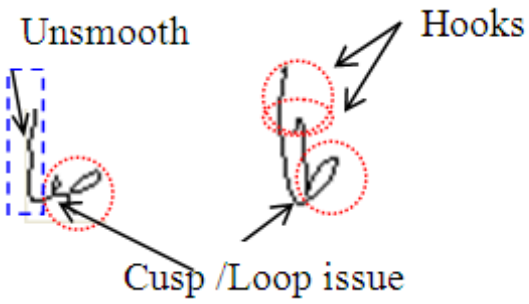

FIGURE 3. Before preprocessing issue (a)صا (b)طا

3.2. **Smoothing and interpolation.** Smoothing is the most significant operation of preprocessing. The main aim of the smoothing is to reduce the variations occurred due to fast writing or natural hand shivering during writing; while unchanging the basic structural of the strokes. In the literature, it is the common operation to normalize the input text. In the case of offline character recognition, an averaging filter is applied on the image, whereas for online, the smoothing filter is only applied to stroke elements array, thus smoothing operation does not affect the width of the stroke elements x, y. Smoothing is performed by averaging the point of length five. Filter is shown below where x_i and y_i are the corresponding coordinates.

$$\text{Filter} = [1/10 \quad 1/5 \quad 2/5 \quad 1/5 \quad 1/10]$$

$$(x_i, y_i) = \frac{(x_{i-2}, y_{i-2})}{10} + \frac{(x_{i-1}, y_{i-1})}{5} + \frac{2(x_{i-5}, y_{i-5})}{5} + \frac{(x_{i1-}, y_{i-1})}{5} + \frac{(x_{i-2}, y_{i-2})}{10}$$

Intermediate points are skipped by pen due to the fast speed of writing and low processing power of pen. Moreover, the points are not equidistant in stroke element but they are equidistant in time. The number of points varies depending upon the writing speed whereas the character recognition system requires equidistant stroke elements. Figure 4 shows the missing points and issues. This missing and un-equidistant data may create some problems in the feature extraction, especially in the case of loop and cusp detection and. N. Mezghani et al. presented an approach for re-sampling the points at equal distance [23]. The intermediate points are estimated after fixed distance, whereas in loop extraction, some problems may occur when the intersection point is not same. Computed the missing data between two points using the interpolation through the Bresenham's line drawing algorithms, shown in Figure 4.

(a) (b)

FIGURE 4. (a) Missing points due to fast speed of writing; (b) smoothed and interpolated stroke

3.3. **Slant estimation.** Slant correction reduces the vertical variation in the handwritten text and it is crucial part for feature extraction, i.e., structural features are highly based on slant and skew corrected strokes. In Urdu script, especially Nasta'liq style, the feature extraction and segmentation is highly dependent on the direction of writing, thus vertical movement of pen must be perpendicular as it is possible, instead of diagonally slanted. Otherwise, slanted strokes resulting into poor features matrix or into bad character segmentation, cannot be performed correctly. For example, for implicit segmentation it is very difficult to adjust the criteria for segmenting the stroke into subunits, i.e., the input stroke is divided into uniform subunits for HMM based recognition. Thus for accurate splitting, the strokes must be vertically perpendicular.

Generally, handwritten strokes are not uniform, thus the slant may change with words even it may be different within ligatures or strokes. Therefore, the global uniform slant estimation is not guaranteed to provide good slant correction for handwritten text. Due to the complexity of Urdu handwritten script, locally slant correction is required instead of global slant correction. Whereas it is not easy to estimate the slant on stroke bases due to very less information for estimation. This problem mostly occur in small (in height and width) strokes, i.e., ﻭ ﺩ ﺭ due to the short information that helps in slant estimation. We normalize the stroke by estimating the slant locally based on the current stroke and some clue from the previous stroke. We compute the slant locally based on vertical projection with the help of neighbor strokes information.

The slant estimation is performed on only the primary strokes because the secondary strokes are slanted naturally due to small size whereas alignment correction is performed on both primary and secondary strokes as shown in Figure 5. For slant estimation, both vertical and horizontal movements of pen are analyzed while slant correction is performed on only vertical axis. As the Urdu ligatures are written from top to bottom and left to right and the writers may write the ligatures little right or left slanted shown in Figure 5. For slant estimation, gravity fall of topmost strokes elements is used with additionally diagonally vertical up or vertical down movement. The slant is corrected by vertical estimation of line from lowest measure towards new upwards estimated points shown in Figure 6. If the topmost point angle with starting point angle is greater than 70 then no slant correction is performed. Otherwise slant correction is performed by using the gravity fall from estimated position to the lowest position and the next segment is attached with the vertically dropped position as shown in Figure 6.

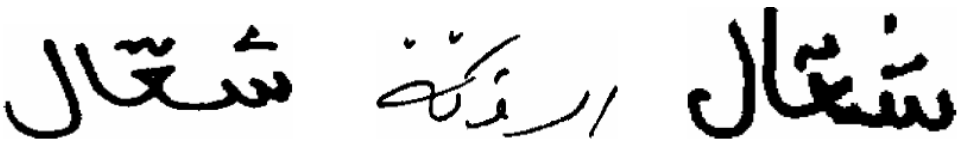

FIGURE 5. Left slanted, right slanted and normal words

FIGURE 6. Locally angle computation for slant normalization

3.4. **Stroke mapping.** It is difficult to write some ligatures i.e., بصر, لا etc. shown in Figure 7. Without lifting the pen as shown in Figure 7, whereas online character recognition does not permit to lift pen during writing of ligature for Urdu script based languages. Thus, to overcome this issue, an algorithm is proposed based on two rules:

$$\theta = \tan^{-1}\left(\sqrt{\frac{(x-a)^2}{(x-a)^2+(y-b)^2}}\right)$$

if $\theta \leq \alpha \,|(x', y') = (x, b)$ and $(x'_i, y'_i) = (x_i, b)$ For all intermediate point i.

Rule 1: Two consecutive ligatures are considered as one ligature if they overlap each other and if the starting point of the second ligature is close to the ending of the first stroke and one of the strokes is vertical ending. This threshold value β is considered twice in vertical than in horizontal.

Rule 2: Two consecutive strokes are combined to form one stroke if the previous strokes ending point is very close to the start of the second stroke; these two ligatures are combined and considered as a single ligature as shown in Figure 8.

FIGURE 7. Few strokes that are difficult to write in a single stroke

(a) (b)

FIGURE 8. (a) Resolved problem shown in Figure 7; (b) error in combining

If

$$dist|(x, y), (m, n)| < \vartheta$$

And both strokes ends at vertically then both strokes are combined

$$dist|(x, y), (m_i, n_i)| < \phi$$

where $i = 1, \cdots, \alpha$, α is depends upon the size of the strokes. The issue in Figure 8(b) can be resolved by using the word level processing.

Delayed strokes (secondary strokes) are necessary part to differentiate the similar strokes in inter languages, i.e., Arabic, Urdu and Persian and also in intra language, i.e., ٹ, ڀ, ڀ. Secondary strokes handling is very important part for the classification of Arabic script based languages. Generally Arabic script based languages contain zero or more secondary strokes corresponding to one primary stroke and smaller in size as compared to primary strokes. The separation of secondary strokes is not easy task because the dots may not appear exactly above or below the character in the stroke and may occur with different order. Furthermore, some delayed strokes, have same or similar shapes as of primary strokes, i.e., "ط" may appear as primary strokes and secondary stroke as well. The localization of diacritical marks is also very critical especially when related character is of very small size or related stroke may contain more than one secondary stroke shown in

Figure 9. It is very difficult to decide that, either the secondary stroke belongs to one character in the ligature or to multiple characters in the ligatures.

TABLE 1. Distance vs position fuzzy rules

Distance/Position	Up	Down	Inside	Left	Right
Small	MS-1	MS-1	MS-1	MS-1	MS-1
Medium	MS-1	MS-1	MS-1	MS-2	MS-2
Large	MS-2	MS-2	MS-2	NS	MS-3
Very Large	MS-3	NS	NS	NS	NS

TABLE 2. Result vs size fuzzy rules

Size/Score Result	MS-1	MS-2	MS-3
Small	SS	SS	NS
Medium	SS	SS	NS
Large	SS	NS	NS
Very Large	SS	NS	NS

*MS May be Secondary Stroke

**NS not Secondary Stroke

The direct projection is not possible due to the complex structure of the Urdu script shown in Figure 9. The algorithms [15] that worked for Arabic, failed for Urdu language due to the script complexity, to the large number of diacritical marks and to the similarly of the secondary strokes and primary strokes. We proposed an algorithm based on fuzzy logics to handle these delayed-strokes through the vertically projection of the stroke to find its corresponding character surrounded by other characters. We consider the vertical projection along with the stroke order information and the candidate primary stroke shape. As in Urdu script, mostly secondary strokes are written above or below the word and in some cases they may appear little before, little after (closely touching the right or left side respectively), or within the word-part with respect to the horizontal axis, as shown in Figure 9. The strokes smaller than the size θ (this letter is reserved to angles, don't use it for sizes) are considered as candidate for the secondary strokes. If the size is greater than θ then the candidate stroke location is projected on to the primary stroke. It is considered as a candidate for the secondary strokes if firstly it lies little up, secondly stroke ending must be from right to left and thirdly its ending point lies within the range $\theta 1$. By combining the fuzzy terms through the logical operators (OR, AND) on fuzzy variables, i.e., location, timing, stroke size, etc., a number of fuzzy rules are constructed for delayed stroke segmentation. The fuzzy membership function is explained in Table 1 and Table 2.

The position of the delayed strokes is calculated by vertical projecting the secondary stroke on the primary stroke. The projection is based on the size of the delayed strokes and timing information delayed strokes. The projection is performed by vertically mapping the ending point of the secondary stroke onto the primary stroke. The position of the secondary stroke is calculated onto the primary stroke as shown in Figure 9.

3.5. **Baseline estimation and skew correction.** The most significant operation in preprocessing is the estimation of the baseline. Baseline is the virtual line on which characters are combined to form the ligatures and it is the necessary requirement for both readers and writers. Without baseline detection it is very difficult and complex to read

the text even for human, and error rate increases up to 10% the context sensitive interpretation is involved. Whereas in automatic classification no context based interpretation is involved, thus baseline detection is the necessary part of better classification especially for Arabic script based languages.

Figure 11 shows baseline and two descender lines. Ascenders and descenders may overlaps with characters part in the main part of the word (central band). To increase

TABLE 3. Diacritical marks separation algorithm

Algorithms: Delayed Stroke Segmentation

For All P, Pi+j
Pi+j is the delayed Stroke, Pi is the concerned primary stroke.
RMER(Pi) is the point on right side of minimum enclosing rectangle
LMER(Pi) is the point on left side of minimum enclosing rectangle
IF Pi+j < θ
Pi+j may be the candidate for secondary stroke C(Sj Pi)
Else
IF| Pi+j (xf, yf) – RMER(Pi) < β) OR | Pi+j (xf, yf) – LMER(Pi)< β
Pi+j may be the candidate for secondary stroke C(Sj Pi)
Otherwise
Pi=Pi+1

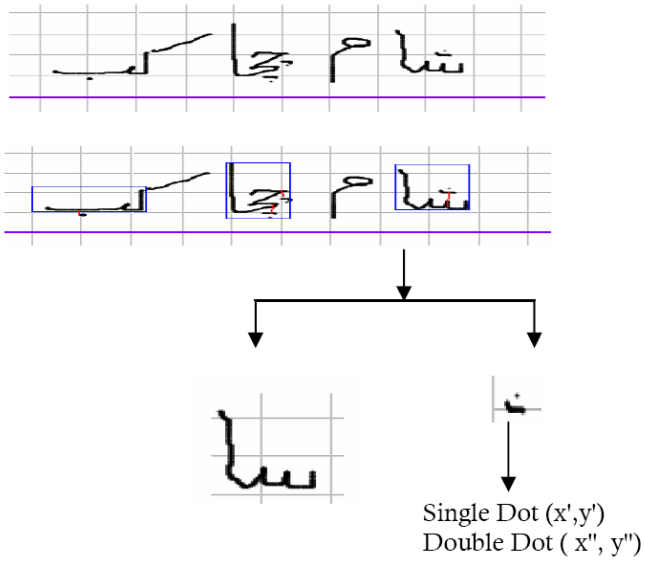

FIGURE 9. Secondary stroke separation

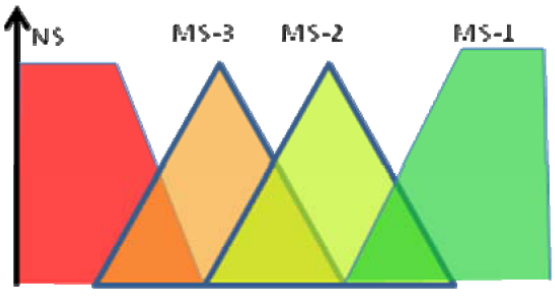

FIGURE 10. Membership function for Table 1 and Table 2

the baseline estimation accuracy, it is necessary to remove the ascenders and descenders before the baseline estimation whereas the correction is performed with accenders and desenders. Different characters appear at different descender lines. Due to the complexity of Nasta'liq over Naskh, one character may appear at different descender lines depending upon the associated characters; whereas in Naskh style, the last character appears on the baseline and does not depend upon its connected character shown in Figure 12. Thus, the baseline estimation for Nasta'liq written text is more complex than Naskh style. Thus, without pre-knowledge on word structure, it is very difficult to estimate the baseline. The blue line in Figure 11 shows the baseline for Nasta'liq and Naskh style. The figure shows that the baseline features are different for Nasta'liq and Naskh.

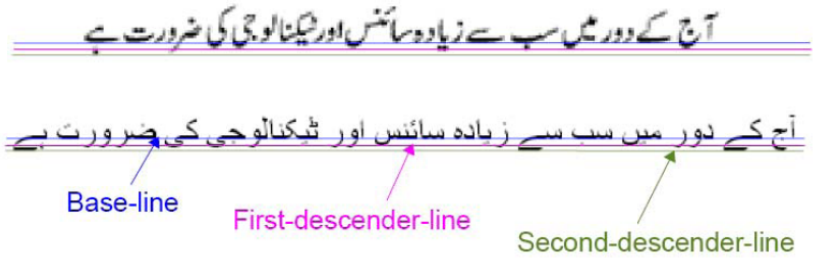

FIGURE 11. Baseline and descender lines for Nasta'liq and Naskh font for Urdu [4]

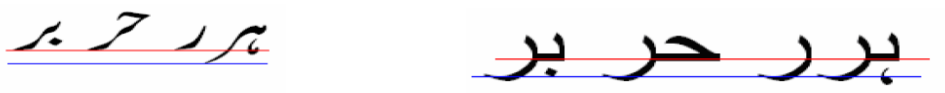

FIGURE 12. Baseline (red) for Nasta'liq and Naskh Style and blue line shows issues in baseline

We present a novel baseline estimation method for only ghost character for online input. The proposed approach is divided into three phases.

- Phase I: Primary baseline estimation.
- Phase II: Locally baseline estimation.
- Phase III: Primary baseline estimation.

Phase I. Primary baseline estimation.

For the primary baseline estimation, we used a projection based method. The primary baseline estimation is only to find a rough baseline for locally baseline estimation. The horizontal projection based method counts the number of elements on horizontal line. The maximum number of elements on horizontal line is the baseline. As the secondary strokes are removed during the phase I, thus the projection baseline is the estimated baseline on ghost character and gives good result by eliminating the influence of diacritical marks on baseline estimation proposed in literature.

Phase II. Locally baseline estimation.

Although the projection based baseline is robust and very easy to estimate, this method requires a long straight text line. Whereas in the case of handwritten text, especially for online handwritten, the line/words length may be very short or very difficult to find a single baseline due to large variation in handwritten text. In the third phase, some features are extracted that helps in baseline detection. These features are used to estimate the baseline locally with the help of primary baseline. The features and locally baseline estimation is fully dependent on the style of script, i.e., on Nasta'liq which has different

set of features with different rules as compared to Naskh. Figure 13 describes the proposed baseline extraction method.

FIGURE 13. (a) Raw input strokes; (b) ghost shapes after separation of secondary strokes; (c) primary baseline estimation based on projection; (d) locally baseline estimation based on features

Features lieing on the baseline are extracted, i.e., ray, bey, etc. shown in Figure 14. As in Nasta'liq the last character lies on the baseline, thus for Nasta'liq the last shapes of the character is extracted for locally baseline estimation. Whereas for Naskh font, local baseline projection is used with some additional features. The baseline is estimated little above the baseline extracted using features for Naskh.

For primary baseline estimation, α is computed based on the horizontal projection as shown in Figure 13(c) on ghost character. The primary baseline estimation is used to reduce the error occurred by using the feature based approach. Then, the second baseline is computed based on the features and pre baseline. The role of the primary baseline is to compute the exact angle for local baseline by using the following relation.

For each ligature: If $|\alpha - \beta_i| < \theta$, then the estimated angle is β_i, else the estimated angle is α, where β is the locally computed angle of each ligature.

The skewness is performed on both ghost strokes and secondary strokes. The angle of secondary strokes is the same angle of associated ghost stroke performed using the following relation.

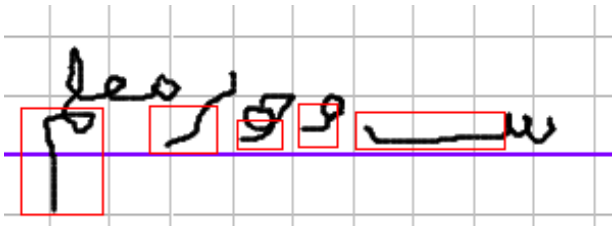

FIGURE 14. Features for baseline estimation

For ghost strokes.

$$x' = x\cos\beta_i - y\sin\beta_i$$
$$y' = y\cos\beta_i - x\cos\beta_i$$

For secondary strokes.

For testing the purpose, we used a character recognition system on a dictionary of size 1800 ligatures for Nasta'liq script and 1000 written in Naskh style [16]. The performance evaluation shows that fuzzy based preprocessing is better opportunity to treat with variation and complexity.

4. **Conclusion.** This paper presents the issues of preprocessing steps for handwritten online character recognition from both online and offline side to reduce the variation by normalizing it to increase the efficiency of the recognition system. As Arabic script based character recognition is a very difficult task due to complex language structure so to attain good accuracy pre-processing of the raw input strokes is crucial part. The secondary strokes separation algorithm based on fuzzy logics to handle these delayed-stroke through the vertically projection on the stroke to find its corresponding character surrounded by multiple character. So, we consider the vertical projection along with timing information and corresponding stroke shape. Baseline estimation is one of the most difficult due to the complex nature of the Urdu scripts based languages. We present a novel technique for the baseline estimation for cursive handwritten Urdu script written in Nasta'liq and Naskh styles. Firstly, the secondary strokes are segmented form the raw input strokes. Then primary baseline is extracted using the horizontal projection on ghost shapes. Finally the locally baseline of each ligature is estimated based on features and primary baseline estimation. The presented approach proved significant result improvement due to mixture of locally baseline estimation over globally baseline estimation and reduction of diacritical marks. The proposed method provides accuracy 74.3% and 60.7% for Nasta'liq and Naskh font respectively. For validation of presented preprocessing approach, we tested on Urdu character recognition system [16] which shows considerable improvement 89.2% (1.6) in results.

REFERENCES

[1] F. Bouchareb, M. Bedda and S. Ouchetati, New preprocessing method for handwritten arabic word, *Asian Journal of Information Technology*, pp.609-613, 2006.

[2] S. A. Hussain, F. Anwar and A. Sajad, Online urdu character recognition system, *Proc. of the IAPR Conference on Machine Vision Applications*, Tokyo, Japan, 2007.

[3] M. I. Razzak , S. A. Husain and M. Sher, Rule based online urdu character recognition, *ICIC Express Letters*, vol.4, no.2, pp.571-576, 2010.

[4] S. Malik and S. A. Khan, Urdu online handwriting recognition, *Proc. of the IEEE Symposium on Emerging Technologies*, pp.27-31, 2005.

[5] M. Hussain and M. N. Khan, Urdu character recognition using spatial temporal neural network, *the 9th International Multitopic Conference*, pp.1-5, 2005.

[6] M. Nakai, N. Akira, H. Shimodaira and S. Sagayama, Substroke approach to HMM-based on-line kanji handwriting recognition, *Proc. of the 6th International Conference on Document Analysis and Recognition*, pp.492-495, 2001.

[7] F. Biadsy, J. El-Sana and N. Habash, Online arabic handwriting recognition using hidden markov models, *Proc. of the 10th International Workshop on Frontiers in Handwriting Recognition*, 2006.

[8] D. Douglas and T. Peucker, Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, *The Canadian Cartographer*, vol.10, no.2, pp.112-122, 1973.

[9] R. I. Elanwar, Simultaneous segmentation and recognition of arabic characters in an unconstrained on-line cursive handwritten document, *International Journal of Computer and Information Science and Engineering*, 2007.

[10] S. A. Adeed, C. Higgens and D. Elliman, Recognition of offline handwritten arabic word using hidden markov model approach, *Proc. of the 16 th International Conference on Pattern Recognition*, vol.3, 2002.

[11] H. Al-Rashaideh, Preprocessing phase for arabic word handwritten recognition, *Information Transmissions in Computer Networks*, pp.11-19, 2006.

[12] H. Boubaker, M. Kherallah and A. M. Alimi, New algorithm of straight or curved baseline detection for short arabic handwritten writing, *The 10th International Conference on Document Analysis and Recognitionm*, 2009.

[13] S. S. Maddouri and H. E. Abed, Baseline extraction: Comparison of six methods on IFN/ENIT database, *International Conference on Frontiers in Handwriting Recognitio*, 2008.

[14] F. Shafait, A. Hasan, D. Keysers and T. M. Breuel, Layout analysis of urdu document images, *International Multitopic Conference*, Islamabad, 2009.

[15] F. Farooq, V. Govindaraju and M. Perrone, Preprocessing methods for handwritten Arabic documents, *Proc. of the 8th International Conference on Document Analysis and Recognition*, pp.267-271, 2005.

[16] K. Horio and T. Yamakawa, Handwritten character recognition based on relative position of local features extracted by self-organizing maps, *International Journal of Innovative Computing, Information and Control*, vol.3, no.4, pp.789-798, 2007.

[17] M. I. Razzak, S. A. Husain and M. Sher, HMM and fuzzy logic: A hybrid approach for online urdu script based languages character recognition, *Knowledge-Based System*, vol.23, no.8, 2010.

[18] R. A. Mohamad, L. L. Sulem and C. Mokbel, Combining slanted-frame classifiers for improved HMM-based arabic handwriting recognition, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.31, no.7, pp.1165-1177, 2009.

[19] J. H. AlKhateeb, J. Ren, S. S. Ipson and J. Jiang, Knowledge-based baseline detection and optimal thresholding for words segmentation in efficient pre-processing of handwritten arabic text, *Proc. of the 5th International Conference on Information Technology: New Generations*, 2008.

[20] N. Araki , Y. Konishi and H. Ishigaki, A statistical approach for a handwritten character recognition using bayesian filter, *International Journal of Innovative Computing, Information and Control*, vol.5, no.11(B), pp.4033-4040, 2009.

[21] A. Benoureth, A. Ennaji and M. Sellami, Semi-continuous HMMs with explicit state duration applied to arabic handwritten word recognition, *Pattern Recognition Letters*, vol.29, no.12, 2008.

[22] M. Pechwitz and V. Maergner, HMM based approach for handwritten arabic word recognition using the IFN/ENITdatabase, *Proc. of the International Conference on Document Analysis and Recognition*, Edinburgh, Scotland, pp.890-894, 2003.

[23] N. Mezghani, A. Mitche and M. Cherit, A new representation of shape and its use of high performance in online Arabic character recognition by an associative memory, *International Journal of Document Analysis*, pp.201-210, 2005.

[24] M. Kherallah, F. Bouriand and A. M. Alimi, On-line Arabic handwriting recognition system based on visual encoding and genetic algorithm, *Engineering Applications of Artificial Intelligence*, vol.22, no.1, pp.153-170, 2009.

[25] B. Chang, H. Tsai and P. Yu, Handwritten character recognition using a neuro-fuzzy system, *International Journal of Innovative Computing, Information and Control*, vol.4, no.9, pp.2345-2362, 2008.

[26] K. Kiyota, N. Ezaki and K. Itou, Development of pen based note taking system for person with visually, *International Journal of Innovative Computing, Information and Control*, vol.5, no.3, pp.653-659, 2009.

[27] M. I. Razzak, S. A. Husain and M. Sher, A fuzzy expert system: Biologically inspired multilayered and multilanguage urdu script character recognition, *International Journal of Innovative Computing, Information and Control*, in press.