Form Field Frame Boundary Removal for Form Processing System in Gurmukhi Script

Dharam Veer Sharma[1] Gurpreet Singh Lehal[2]
Department of Computer Science, Punjabi University, Patiala, INDIA
dveer72@hotmail.com[1], gslehal@gmail.com[2]

Abstract

Machine recognition of hand-filled forms is a challenging task. Form processing involves many activities including form field location, field frame boundary removal and data image extraction, segmentation, feature extraction, classification and recognition. The paper proposes an algorithm for removal of the field frame boundary of the hand filled forms in Gurmukhi Script. Because of the structural characteristics of the Gurmukhi script, use of headline and varied writing styles, the filled data may overlap or get merged with the field frame boundaries, which make the field data extraction task very challenging. It becomes particularly difficult to remove the field frame boundaries while preserving the filled in data. Experimental results reveal the efficiency of the proposed method in removing the field frame boundary and extracting the field data from form documents. Though, the algorithm has been developed and tested for Gurmukhi script but with minor or no changes it can be applied to scripts having structural features similar to that of Gurmukhi script, like Bangla and Devnagari.

1. Introduction

Hand-filled Forms are information carriers and frequently used for collecting data from different sources. The collected data is entered in computers for processing. Forms may vary from paper based to online. Manual keying-in data, for processing, requires manpower and is prone to errors. It costs in terms of time and money. However, it will be useful to deploy automated systems for reading data from paper based forms and storing it in a form which can be modified, processed and analyzed. Reading of data from paper based forms requires converting the form data into digital format, which can be recognized and processed by computers. This can be done by feeding the paper forms to a system which recognizes the image of the paper form and converts it into fields consisting of set of characters.

Examination of literature reveals that the recognition of handwritten characters is relatively difficult task, because of the large number of classes, especially if we consider that well formed characters can not be expected in case of common forms. The use of vowels (matras), half vowels, half characters and the line connecting the different characters of a words used in some Indian scripts add to the complexity of recognition manifolds.

The paper has been divided into 7 sections. section 2 discusses some existing work done related with form processing, the proposed algorithm is explain in section 3 and section 4 covers the experimental results and discussions, while conclusions of the study are covered under section 5. Section 6 covers the references used.

2. Previous work

Some well developed systems are available for recognizing and processing data of hand-filled paper forms in European and Oriental languages. There is no work reported for recognition and extraction of hand-filled text from paper based form for Indian scripts.

Frame line detection is the most important and difficult step of form recognition. Hough transform as given by Illingworth, J., et al[12] and vectorization by Wenyin, L., et al[13] are two kinds of widely used line detection methods. As a global approach, Hough transform can detect dashed or broken lines. However, it is too slow to be applied in form recognition. Most of the frame lines on forms are horizontal or vertical and modified Hough transforms are just projection approaches given by Jinhui, l. et at[10] and Jiun, L., et al[11]. Though fast, projection approaches have some problems. First, they cannot detect diagonal lines and

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.179

frame lines with large skew angles. Second, when characters overlap or merge with frame lines, the projection of frame lines are overwhelmed in the projection of characters. Third, some frame lines in a scanned image, especially those on the image borders, are deformed. With some kind of curve, they are not straight. Projection methods fail to detect such curved lines too. As the other kind of algorithms widely used, vectorization approaches of Wenyin, L., et al[13] extract vectors from images first. By merging these vectors, the whole objects are detected. Such bottom-to-up approaches can solve the above problems of projection approaches. For skewed images projection should be performed at the skew angle as suggested by Liu, J., et al[14]. The angle can be estimated according to the slope of the top horizontal frame line, which can be reliably detected.

In their paper Shimamura, T. et al[15] have suggested carrying out erosion several times for removal of field frame lines if frame lines are thinner than the handwritten data. Then dilation can be applied for the same number of times as was erosion applied then the handwritten data will be almost of the same thickness as it was before applying erosion. Application of this approach is practically not possible as handwritten data may be of varying thickness and in some cases it may be thinner than the frame boundaries leading to removal of handwritten data.

Some authors have suggested that in order to remove the box, a set of regions of each edge be extracted and a standard line fitting technique be used to parameterize them as given by Simoncini, L. et al[16]. The deletion of the lines is carried out, leading to an excessive erosion of the crossing strokes. At the end, they must be repaired and the crossing characters reconstructed. Employing this method leads to holes in the image data, in cases, where field data either overlaps or is merged with the frame boundary.

A structure for a form reader whose performance is based on supervised learning has been described by Lam, S.W. et al[1]. The recognition is based on contexts. A system for automating data entry system by recognizing data from forms has been suggested by Lorie, R. A. et al[2], in which, authors have suggested use of contexts in post-processing for improvement of recognition results and involvements of user intervention to verify the results of fields which are difficult to recognize. A design for automated data entry from handwritten forms, which is based on design of a template form given by Ning, L. W. et al[3] for capturing regions of interest only from the forms has also been suggested.

The system proposed by Kavallieratos, E. et al[4] is based on hidden Markov models. After lexical confirmations of the result of recognition achieved were 97%. In a general system for extraction and cleaning of data from handwritten forms given by Ye, X. et al[5], the items of interest are located from the form for which a model template is generated from a blank form, which is used to remove the form frame from the actual forms to be used for recognition. Morphological operations based on statistical features are used to clean the handwriting touching the pre-printed text. A recognition rate of 95.5% has been reported to be achieved. A system named "Name and Address Block Reader (NABR)" exists for reading names and addressed from tax forms of the Internal Revenue Services of United States proposed by Srihari, S.N. et al[6]. The system is capable of recognizing machine-printed as well as hand-printed data. An OCR correct rate of 89.53% and 97.86% for hand-printed and machine-printed data respectively has been achieved. A form reading technology based on form type identification and form-data recognition by Sako, H. et al[7] with recognition rate of 97% has been reported.

In a kind of vectorization algorithm, which uses a novel image structure element named "Directional Single-Connected Chain (DSCC)" given by Zheng, Y. et al[8] as the elementary vector, DSCC bears appropriate size and can be easily stored and processed, in addition to the capability to solve most types of character-line crossing problems. By merging DSCCs under some constraints, most of the frame lines can be detected correctly. However, there may still exist two kinds of misdetection, i.e., the pseudo lines and the broken lines.

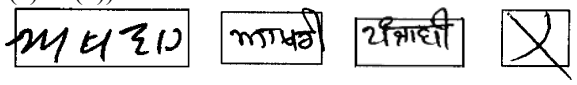
In most frame line detection algorithms, a critical threshold is used to remove any short lines formed by character strokes. This threshold represents the character size. However due to the difference of forms and resolution of scanners, the width or height of characters varies greatly from 10 pixels to more than 100 pixels. In most of the literature, this important threshold is input by users as suggested by Shiyan, P.[9] or is a constant value as proposed by Jinhui, l. et at[10] and Jiun, L., et al[11].

3. Proposed solution

This paper proposes a method which is primarily intended for Gurmukhi script but can be applied to other scripts, like Devnagari and Bangla, having characteristics similar to Gurmukhi script, with no or minor modification.

Most of the forms consist of less than 20% of recognizable region. Applying skew correction on whole of the form image consumes a lot of time as skew correction is performed at pixel level and about 80% of irrelevant pixels are also considered. Time

complexity reduces considerably by considering one field of the form at a time for skew correction. This way skew correction is performed only on the target areas and not on whole of the form image. Having corrected skew of the target field area, the next step is to identify the field bounding rectangle, which may exceed actual width if some left or right (fig. 1. (a) & (b)) line overlapping is there. It may exceed the actual height if some top or bottom overlapping is there (fig. 1 (c) & (d)).



(a) (b) (c) (d)

Figure 1: Examples of overlapping on all four sides

The problem becomes even more complex to handle when the head line of a word is merged with the top frame line as in the case of fig. 2. Gurmukhi script has set of some characters which vary only on the basis of presence or absence of head line e.g. ਸ - ਮ, ਪ - ਧ, ਖ - ਥ. The present algorithm can not help in distinguishing amongst these characters, however in such cases top frame line is successfully removed.

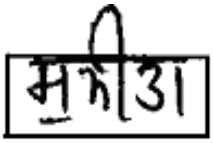


Figure 2: Merger of head line with top line of frame

Figure 3: Wider field area than the filled in data

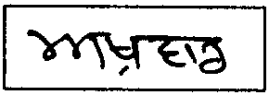


Figure 4: Field data not overlapping frame boundaries.

The proposed algorithm is based on the following assumptions.

i. The form has a starting point in the left top position and an end point in the right bottom position, which forms the basis of field location identification.
ii. Distance between starting point and ending point are stored for calculating any deviation for skew detection during recognition.
iii. Fields are rectangular in shape.
iv. Total number of fields and relative locations (from starting point) of each of the form fields are store as definition of the form, along with data type and constraints on the field.
v. Field frame boundary lines are unbroken.

Using the above mentioned assumptions, field frame boundaries removal is carried out using the following steps:

i. First the starting and ending points in the form are detected. If the distance between these points deviates from the stored distance then the form is skew and skew angle is calculated.
ii. Using the location of starting point, relative location of fields from the starting point and skew angle, field frames are detected.
iii. Skewness is corrected for the field using the calculated skew angle.
iv. From the field frames, LeftTop, LeftBottom, TopRight and BottomRight points are detected.
v. If the height and/or width of the field frame exceed the pre-defined size of the field then some part of the data is overlapping the field frame as in all cases of fig. 1.
vi. If no overlapping of filled data with frame boundaries is detected as in step v (fig. 4), then calculate the horizontal and vertical histogram values of the field frame. From the histogram values find the areas on each side which have histogram values less than or equal to the threshold value and remove all such areas. The threshold value is calculated by doubling the value of frame line thickness (for two sides of the bounding rectangle).
vii. For overlapped fields, contours are traced from inside and outside the frame boundary lines, to detect the sides on which the data is overlapping or merged with the frame boundaries. If any junction, as per fig. 5 is detected then the data is overlapping the frame boundaries.
viii. For overlapping fields the lines on sides, where the data is overlapping the frame boundaries are identified and all other lines where no overlapping is encountered are removed using step vi.
ix. For lines with overlapping data the points calculated in step vii are used for tracing and removing lines. While tracing, wherever overlapping is encountered at a point, the junction are located and if a junction is like any of those given in fig. 5 then such points of line are not removed, as this may lead to breaking of characters.

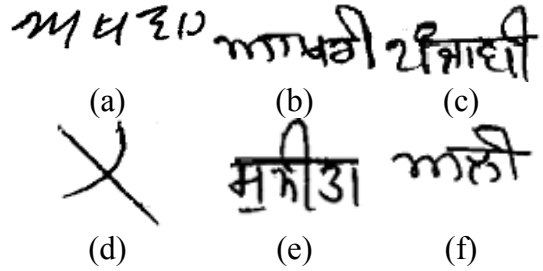
a b c d e

Figure 5: Possible set of junctions at a point

When the field data is merged with the top frame line, as in fig. 2, the top frame line is removed from the sides of the word by calculating the vertical projections and the merged line is not touched as this may lead to erroneous removal of headline of some characters.

4. Experimental results

In absence of any bench mark image database the algorithm was tested on a total of 200 forms of same type. Each form consisted of 42 fields. The forms were filled by different persons with their natural handwriting. The average time required for field detection, skew correction and field frame removal was 2.4 seconds for a form containing 42 fields scanned under 300 resolution as bi-level images on a PIV 2.66 Ghz, 1 GB system. Of the total 8400 fields the algorithm correctly removed frame boundary of 8326 fields which is 99.12 percent. No breaking of characters or loss of significant data was witnessed. Fig. 6 shows some of the results obtained by applying the algorithm on the field data of figs. 1, 2 and 3 above.



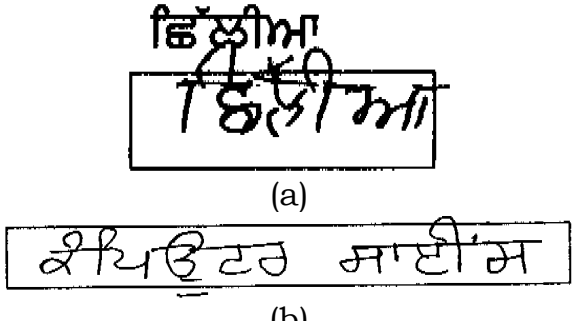
(a) (b) (c)

(d) (e) (f)

Figure 6: Results of field frame boundary removal

The system fails to remove field boundary under the following cases:

i. Where field data overlaps or is merged with the field caption placed on top of the field as in fig. 7(a).
ii. When an isolated part of the word lies outside the field frame boundary as in fig. 7 (b).

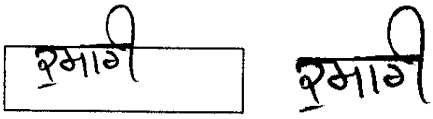


(a)

(b)

Figure 7: Some failure cases of field frame boundary removal

The limitation of this algorithm is that if the word of the field contains a character without headline, like ਖ , ਪ , ਮ , ਘ etc. then headline is added to such characters as well, as shown in fig. 8(a). The actual word, in figure 8 (a) is ਕੁਮਾਰੀ but converted to ਕੁਸਾਰੀ after form field frame boundary removal as in fig. 8(b).



(a) (b)

Figure 8: Merger of field frame boundary with word headline

5. Conclusion

In the present paper, a novel approach has been proposed to remove the form field frame boundary, while preserving the data contained therein. The method does not result in any breaking of characters and thus improving the recognition results. Experimental results reveal the feasibility and efficiency of the proposed approach in form recognition. Future research will develop more efficient recognition algorithm which can incorporates heuristics to improve the recognition speed and form tolerance.

6. References

[1] S.W. Lam, L. Javanbakht, S.N. Srihari, "Anatomy of a Form Reader", *Proc. of 2nd ICDAR*, Tsukuba Science City, Japan, 1993, pp. 506-509.

[2] R. A. Lorie, V. P. Riyaz, T. K. Truong, "A System for Automated Data Entry From Forms", *Proc. of 13th ICPR*, Vienna, Austria, 1996, vol. 3, pp. 686-690.

[3] L. W. Ning, Y. K. Siah, M. Khalid, M. Yusof, "Design of an Automated Data Entry System for Hand-filled Forms", *Proc. of TENCON*, 2000, vol. 1, pp. 162-166.

[4] E. Kavallieratos, N. Antoniades, N. Fakotakis, G. Kokkinakis, "Extraction and Recognition of Handwritten Alphanumeric Characters From Application Forms", *Proc. of 13th Int. Conf. on Digital Signal Processing*, 1997, vol. 2, pp. 695-698.

[5] X. Ye, M. Cheriet, C. Y. Suen, "A Generic system to Extract and Clean Handwritten Data from Business Forms", *Proc. of 7th IWFHR*, Amsterdam, Netherlands, 2000, pp 63-72.

[6] S.N. Srihari, Y. C. Shin, V. Ramanaprasad, D. S. Lee, "A System to Read Names and Addresses on Tax Forms", *Proc. of the IEEE*, 1996, vol. 84, issue. 7, pp. 1038-1049.

[7] H. Sako, M. Seki, N. Furukawa, H. Ikeda, A. Imaizumi, "Form Reading based on Form-type Identification and Form-data Recognition", *Proc. of 7th ICDAR*, Edinburgh, Scotland, 2003, pp. 926-930.

[8] Y. Zheng, C. Liu, X. Ding, S. Pan, "Form Frame Line Detection with Directional Single-Connected Chain", *Proc. of 6th ICDAR*, Seattle, USA, 2001, pp. 699-704.

[9] Pan Shiyan, "Research and Realization of a General Form Recognition System", *Master thesis of Tsinghua University*, 1999.

[10] Liu Jinhui, Ding Xiaoqing, Wu Youshou, "Description and Recognition of Form and Automated Form Data Entry", *Proc. of 3rd ICDAR*, Montreal, Canada, 1995, pp. 579-582.

[11] Jiun-Lin Chen, Hsi-Jian Lee, "An Efficient Algorithm for Form Structure Extraction Using Strip Projection", *Pattern Recognition*, 1998, Vol.31, No.9, pp.1353-1368.

[12] J. Illingworth, J. Kittler, "A Survey of the Hough Transform", *Computer Vision, Graphics, & Image Processing*, 1988 vol.44, pp.87-116.

[13] Wenyin Liu, Dov Dori, "From Raster to Vectors: Extracting Visual Information from Line Drawings", *Pattern Analysis & Application*, 1999, No.2, pp.10-21.

[14] J. Liu, X. Ding, Y. Wu, "Description and Recognition of Form and Automated Form Data Entry", *Proc. of 3rd ICDAR*, Montreal, Canada,, 1995, pp. 579-582.

[15] T. Shimamura, B. Zhu, A. Masuda, M. Onuma, T. Sakurada, M. Nakagawa, "A Prototype of an Active Form System", *Proc. of 7th ICDAR*, Edinburgh, Scotland, 2003, vol. 2, pp. 921-926.

[16] L. Simoncini, V. Kovbcs, M. Zs., "A System for Reading USA Census '90 Hand-Written Fields", *Proc. of 3rd Int. Conf. on Document Analysis and Recognition*, Montreal, Canada, 1995, vol. 1, pp. 86-90.