

Cell extraction and horizontal scale correction in structured documents

In a structured document like forms, cheques, etc., there is a predefined space for the user to fill an entry called frame field. When the user is writing he/she makes an attempt to accommodate the handwritten content within the form field. This leads to nonuniform intercharacter spacing within certain words where the starting characters of the word are written with normal spacing and the latter/towards the end characters are written in compact form so as to accommodate the word with the frame field. In this chapter, we extract the layout of the structured form images and perform horizontal scale normalization of the text written inside the frame fields.

This chapter is organized as follows. Section 4.1 describes the problem and its significance. In Section 4.2 we discuss the related work for cell extraction and text normalization. Section 4.3 focuses on the proposed methodology. Section 4.4 explains the experimental section which includes proposed dataset and implementation details. 4.5 indicates the performance of the proposed method compared to other state-of-the-art methods. Section 4.6 presents the concluding remarks.

4.1 INTRODUCTION

Offline handwritten text recognition is a challenging task even after recent advances in the technology. There are various challenges involved in this task, major of all is writer variability. Other challenges as mentioned in (Bal and Saha [2018]; Brink et al. [2011]; Uchida et al. [2001]; Singh et al. [2008]) include degradation in the text, slant, skew, etc. While considering structured documents like forms, cheques, etc., there arise other challenges (Sharma and Lehal [2009]; Chen and Lee [2001]; Xi and Lee [2005]) like localizing handwritten text among printed text, text crossing their respective field boundary, etc. Pre-processing steps such as slant and skew correction have significant role in increasing the efficiency of handwritten text recognition. They reduce inter and intra writer variability leading to increased recognition accuracy.

A form can be considered as a structured document composed of three main elements, viz., horizontal and vertical layout lines, preprinted data, and user filled-in data. A Field frame is a predefined space for the user to fill the entry (Perez-Cortes et al. [2008]). When a user writes within a form field it is often seen that the starting characters of the word are written in a sparse manner but towards the end, it gets squeezed to a compact form as the user attempts to accommodate the word/sentence within the frame field. For applications like word spotting and word recognition, this can affect the performance. Hence the variation in horizontal scale needs to be corrected. This paper addresses the problem of cell extraction and horizontal scale correction of words written in form fields. Datasets for this task are not available, so we have created our own dataset comprising 200 words obtained from multiwriter bilingual (English and Devanagari) forms of 10 different layouts with 10 forms of each layout capturing a variety of within word inter-character spacings.

4.2 RELATED WORK

Related work is divided into two subsections. The first part briefs about the form frame lines and cell extraction while the second part discusses various text normalization techniques.

4.2.1 Cell Extraction

A form consists of various frame lines which form boundaries of cells in which the user fills the desired text. These lines exhibit structural layout of the form document. The data filled in a cell have well defined semantics. In (Hori and Doermann [1995]) a method based on Box-Driven Reasoning(BDR) is developed for structural analysis in table-form documents. The method recognizes cells in tabular documents with broken lines and touching characters. Extracting these frame lines is also useful in classification and recognition of the form. In an organization, similar forms have similar layout. Extracting the form layout and exploring them can lead to their application in form classification and recognition. Hierarchical representation of various cells forming the structure of form documents is proposed in (Duygulu and Atalay [2002]) for identification and retrieval of forms. A heuristic algorithm based on the XY-tree method (Nagy and Seth [1984]; Cesarini et al. [1999]) is used to transform geometric structure of the form document to a hierarchical structure by using the horizontal and vertical layout lines. The hierarchical structure of the tree corresponds to the hierarchy of the blocks of the form document. This representation can handle the minor variations of the form in the retrieval process.

4.2.2 Text Normalization

Text normalization is an important preprocessing step in the recognition system. Text normalization is of two types, vertical and horizontal text normalization, corresponding to the word being scaled in the vertical and horizontal direction, respectively. In the literature, various approaches for vertical normalization are proposed. (Gorbe-Moya et al. [2008]) uses a supervised learning method, a method based on HMM/ANN (Pastor-Pellicer et al. [2014]) is proposed for short sentences or isolated words.

A technique for horizontal normalization is proposed in (Marti and Bunke [2001]) where the number of black-white transitions in the horizontal direction are determined for each image line. Using the maximum number of transitions and the image width, the mean number of strokes per pixel is estimated. This number is set in relation to a reference value for the mean number which is computed off-line by statistics over the set of all the text lines in the image database. A scaling factor for horizontal normalization is set according to the relation between the actual and the reference value calculated by the above mentioned approach. In this approach uniform horizontal normalization is carried out over the whole text line. This same method is adopted in (Graves et al. [2009]; Zamora-Martinez et al. [2014]) for horizontal normalization but for the word and not on complete text lines. These methods provide uniform horizontal normalization such that the width of the text lines as in (Marti and Bunke [2001]) and width of the word (Graves et al. [2009]; Zamora-Martinez et al. [2014]) is of the same size depending on the scaling factor. This method does not resolve the issue of the non-uniformity of inter-character space between characters within a word. To address this issue a method that applies non-uniform scaling is required.

The proposed method addresses the issue of inconsistent spacing between characters within a word. This scenario is common in handwritten words, however, more specifically when the handwritten text is written in frame fields. In such cases, characters are sparsely written at the beginning of the word and compactly written towards the end of the word to accommodate text in frame fields. In our approach interest points are marked to locate the sparsely and densely written part. Non-uniform scaling is used such that only horizontal strokes are scaled with least distortions to vertical strokes. The next section describes the proposed methodology.

[illegible]

a **b**

The proposed methodology finds application in structured documents such as forms. Before proceeding to scale correction of handwritten word images we need to locate and extract the handwritten words in a form. Our workflow comprises two main steps, namely, cell extraction, and horizontal scale correction. Using the method proposed in (Espana-Boquera et al. [2011]) and modifying the region growing algorithm, form layout is identified for cell extraction. We make the assumption that out of the printed and handwritten words located in various frames of the form, we know the location of the frames that have handwritten text written in it. So the scale correction is performed directly on the handwritten words.

The method given by (Akinlar and Topal [2011]) is used to detect and extract all the line segments present in an image. In the case of a structured document, it returns the line segments which form the cell boundaries. However, since the algorithm is very sensitive, all the line segments including straight strokes of characters and shirorekha in the case of Devanagari, are also extracted as a noise. During digitization, the document image gets degraded and the cell boundary is distorted. It leads to extraction of the line segment in forms but with noise and distorted cell boundaries. Figure 4.1a shows a sample input form along with the line segments extracted in Figure 4.1b using the method proposed in (Akinlar and Topal [2011]). The enlarged view of a part of the form as shown in Figure 4.2 shows the distorted cell boundaries termed as leaky gaps and extra line segments as a noise.

39

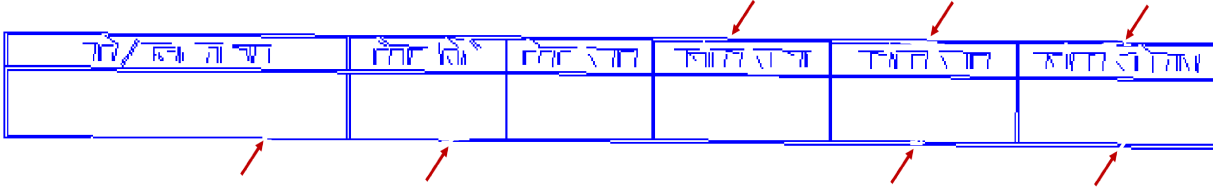


Figure 4.2 : Enlarged image of a part of Figure 4.1b showing noisy gaps (arrows) in frame lines.

(Akinlar and Topal [2011]) noisy gaps present in the frame due to broken lines allows the region to spill out from those gaps. As a result, cells get merged and more than one cell forms a single region. This happens because for a pixel of some predefined type (background pixels in our case), if any of its 8 connected neighbours belongs to its type the region expands through that pixel. In our modified approach we impose the condition that, if the number of pixels determined by constant α multiplied with 8 connected neighbours belongs to background pixels then the adjacent pixel is set to a new seed point and the region expands. The value of α is found to be 0.9 experimentally. Let the current seed point be denoted as $g(x,y)$, the next seed point as $h(x,y)$, the $P(BG)$ be the number of pixels belonging to Background pixels in the 8 connected neighbourhood.

The mathematical expression for modified region growing is:

$$h(x,y) = \begin{cases} BG, & \text{if } P(BG) \geq \alpha \times 8 \\ FG, & \text{otherwise.} \end{cases} \quad (4.1)$$

This criteria prevents the region growing process from leaking through noisy gaps in the lines such that all the cells of the frames are extracted as separate regions. The steps for our proposed modified region growing algorithm is illustrated in algorithm 1. Figure 4.3 shows cell extracted from original and modified region growing method.

Algorithm 1 Modified Region Growing

- 1: Input : Form Image after Line Extraction
 - 2: Initialize a *queue*
 - 3: Initialize Flag=0
 - 4: Initialize seed location such as all 8 neighbour belong to background
 - 5: while (Queue is not Empty and All the pixels of image is traversed) do
 - 6: if image(i,j) is seed then
 - 7: Mark Seed location in Flag as 2
 - 8: Mask = 8 connected neighbour of seed
 - 9: if 90% of Mask is background then
 - 10: Push 8 connected neighbours in queue
 - 11: Mark 8 connected neighbour in Flag as 1
 - 12: NewSeed := pop queue
 - 13: end if
 - 14: end if
 - 15: end while
 - 16: Return Flag
-

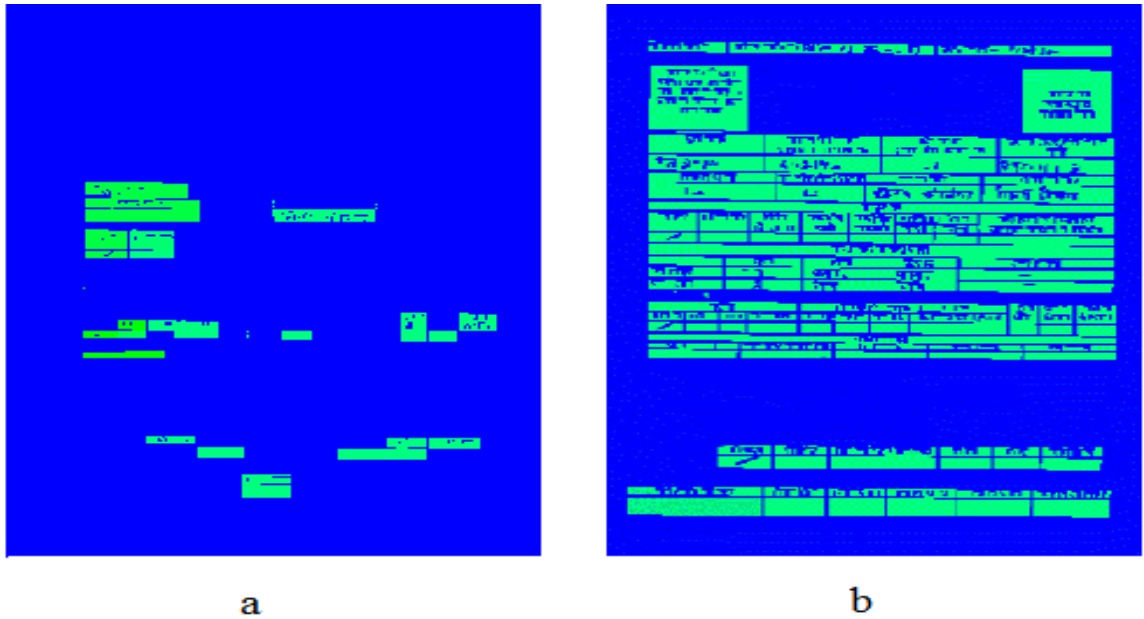


Figure 4.3 : a. Cell Extraction using the Region Growing approach in (Akinlar and Topal [2011]). b. Cell Extraction using the proposed Modified Region Growing approach

4.3.2 Horizontal Scale Correction

After extracting the cell, its content i.e., the handwritten word is normalized so that the subsequent tasks such as word spotting, word recognition can be carried out efficiently. Inconsistent spacing within word leading to variations in horizontal scale needs to be corrected. Essentially there will be need to shrink the sparsely written text and stretch the compactly written text within a word, ensuring that only the white space within the word is scaled, i.e., only horizontal scale correction is done so that vertical strokes are minimally blurred.

We need to scale the word in such a manner that scaling is done at locations where the inter-character spacings are small. These locations where scaling is required are called key locations. We need to find these locations and for this purpose, interest points are computed. These interest points in a word are identified by extracting local minima of the lower contour of a word.

Scaling is applied to the area between the local minima using a window-based approach. This window has a height same as that of the image and a width equal to the distance between consecutive local minima. This results in windows that can have variable width. The scaling factor is calculated from the relationship between actual distance and reference distance between the interest points. The reference distance is computed by taking the mean of all the distances between adjacent local minima. Horizontal Scale correction inside a window for a given scale factor is done by the standard bicubic interpolation method.

While performing horizontal scale correction, the vertical stroke should not be distorted. The local minima in a word image can correspond to the lowest coordinate of a vertical stroke or a horizontal stroke joining two characters or can be within a character. In a word image, each stroke has some thickness. Therefore for vertical strokes, the location of the minimum will extend to its thickness. Considering the starting index of local minimum of vertical stroke will lead to expansion or shrinkage of vertical stroke when scaled to the next local minimum. Considering the last index of local minimum of vertical stroke will also lead to this distortion when scaled to the previous local

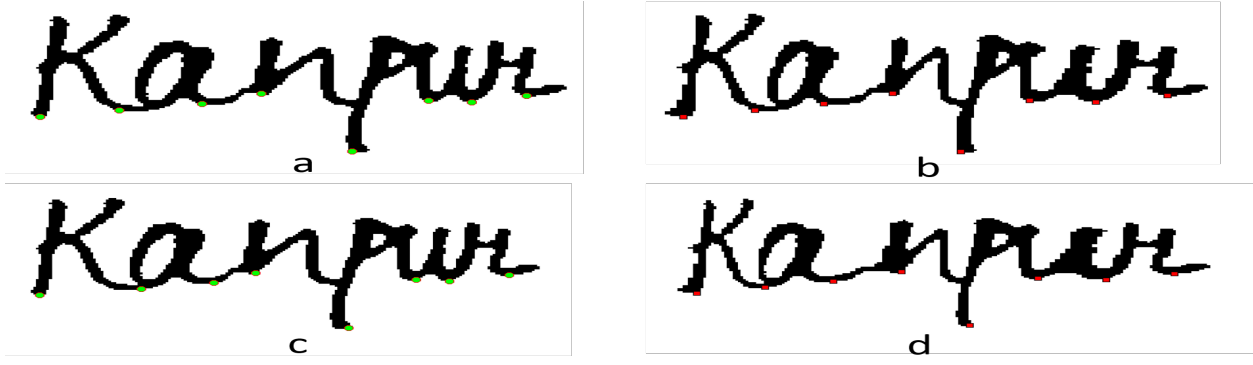


Figure 4.4 : a. Input image with starting index of local minima marked with green dots. b. Horizontal scaled corrected obtained when starting index of local minima is considered. c. Input image with last index of local minima marked with asterisk. d. Horizontal scaled corrected obtained when last index of local minima is considered.

minimum. Both the cases where the starting index and where the last index of local minimum is considered are shown in Figure 4.4. For this purpose the starting index of local minimum is taken as a key location when scaling is done with the previous key location and end index is taken when scaling is done with next location. Hence, selective scaling is applied between these local minima and not for the width of local minimum. This ensures that only horizontal strokes are scaled and vertical strokes are least distorted during the process. The complete procedure for horizontal scale correction is described in algorithm 2. Figure 4.5 shows the width of the vertical stroke. Horizontal lines show the region where scale correction will be applied.

Algorithm 2 Horizontal Scale Correction

```

1: Input : Word Image
2: for i := 1 to NumCols do
3:   LowerContour := Last Foreground Pixel of  $i^{th}$  column
4: end for
5: InterestPoints := LocalMinima(LowerContour)
6: ScaleFactor := Mean(distance between adjacent Interest Points)
7: StartPos := 1
8: for Pos := 1 to NumOfInterestPoints do do
9:   EndPos := InterestPoints(Pos)
10:  Out := Interpolate(WordImage(AllRows, StartPos : EndPos))
11:  OutImage := Append(OutImage, Out)
12:  StartPos=EndPos+1
13: end for
14: Return OutImage

```

4.4 EXPERIMENT

We start with describing the dataset used for experiments. This is followed by a discussion on the implementation details. Finally, the efficiency of our system is described in the result section.

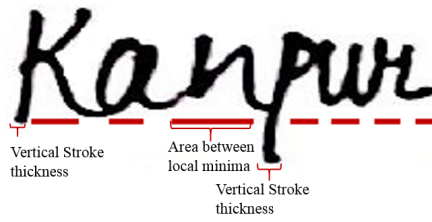


Figure 4.5 : Width of Vertical Strokes and Selective scaling windows between location minima.

Table 4.1 : Sample images of our Form dataset.

4.4.1 Dataset

The proposed approach is tested on two scripts, namely, English and Devanagari. A dataset for this specific problem is not available, so we have created our dataset comprising 200 words obtained from bilingual (English and Devanagari Script) forms of 10 different layouts with 10 forms of each layout written by 10 writers to incorporate inter writer variability to our dataset. Various inter-character space variations are made at different locations in our dataset. Sample images from our form dataset are shown in table 4.1. The method for word recognition in (Almazán et al. [2014]) for the English script is adopted to validate our work. For training the recognition system (Almazán et al. [2014]) the IAM dataset (Marti and Bunke [2002]) is used. The system is trained on the IAM dataset and then tested on our images of English script. For the purpose of training 1000 images of IAM are used and 100 images of IAM and 100 images of our dataset are used for testing. Figure 4.6 shows sample images of our word dataset.

भासिक जनसामान्य संचारिक गवर्नर उम्मीदों
परिक्षा शास्त्री नगर राजनीति अनुमान अंक
Journal Technology Examination राजस्थान
Communication Determination Test situation

Figure 4.6 : Sample images from our scale correction word dataset

4.4.2 Implementation

The handwritten text after horizontal normalization is fed into the word recognition system proposed in (Almazán et al. [2014]). It verifies the utility of our framework in the recognition task. A quantitative and qualitative analysis is performed on the results. The recognition model (Almazán et al. [2014]) works only on the English script, therefore, quantitative analysis is made for the English script only whereas qualitative results are shown for both English and Devanagari scripts. The word recognition method is trained with IAM images. Testing is done using 100 IAM and 100 English words images from our dataset.

Three types of experiments are performed for quantitative analysis. Firstly, without applying horizontal normalization, word recognition is carried out using the method given in (Almazán et al. [2014]). Next, the horizontal normalization proposed in (Marti and Bunke [2001]) is performed on testing images and then the recognition of these normalized images is done. Then the same procedure is repeated but with, normalization performed by the proposed approach. Experimental conditions are set similar to those mentioned in (Almazán et al. [2014]). To evaluate the performance, Character Error Rate (CER) is calculated for each system. The CER between two words is defined as the edit distance or Levenshtein distance between them. It signifies the minimum number of character insertions, deletions, and substitutions needed to transform one word into the other, normalized by the length of the words.

The qualitative analysis of result is done for word images of both English and Devanagari script. The method proposed in (Marti and Bunke [2001]) for horizontal normalization is taken as a baseline method.

4.5 RESULTS AND DISCUSSIONS

Table 4.2 illustrates the results obtained from the word recognition task. The CER metric is used for comparison where the lower score is better. The CER metric is compared for the baseline method and for the method that does not use any horizontal normalization. Results prove that our horizontal scale normalization step reduces the error rate in comparison to other approaches.

Qualitative results of horizontal scale correction are shown in Figure 4.7. The first row corresponds to the input image, the middle row for results obtained by applying the method proposed in (Marti and Bunke [2001]) and the last row shows the resultant images obtained by applying our proposed method. Figure 4.7a shows results for English script and Figure 4.7b for Devanagari script. Red asterisk depicts the interest point locations. It can be seen that in the first and middle row the consecutive distance between adjacent interest points is unequal which is normalized in the last row. In the method proposed in (Marti and Bunke [2001]), the maximum

Table 4.2 : Text Recognition Error Rates

Method	CER (IAM)	CER (Proposed Dataset)
Without Normalization (Almazán et al. [2014])	34.91	32.11
Using Normalization (Marti and Bunke [2001])	30.57	29.83
Using proposed Normalization	38.43	26.32

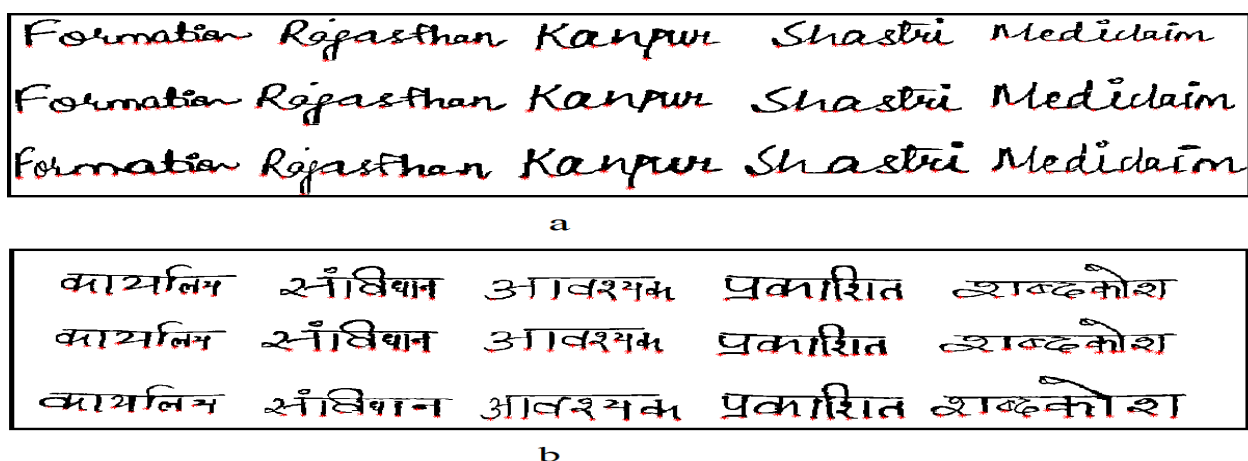


Figure 4.7 : a. Horizontal Scale Correction on English script. b. Horizontal Scale Correction of Devanagari Script. Top row signifies the input word images, middle row the result obtained using the approach of (Marti and Bunke [2001]) and bottom row shows the result from our approach. Red asterisk denotes the interest points.

number of black to white transitions along the various text lines is considered which signifies the vertical strokes present in a word. This method has two shortcomings, firstly, the maximum number of transitions lead to unnecessary number of strokes. Due to writer variability the number of strokes for a word can vary among writers. The second shortcoming is that they are using uniform scaling, which does not have the effect on scale correction within a word. The proposed method eliminates these by using non-uniform scale correction using the local minima of the lower contour of a word. As can be seen in Figure 4.7 that in an upper and middle row these interest points are at unequal distance to each other and in the bottom row all the interest points are at a constant distance to their adjacent interest points.

4.6 CONCLUSION

Preprocessing steps in document analysis are crucial for achieving increased accuracy for many important tasks like handwritten text recognition, word spotting, etc. In this chapter, we proposed a method for horizontal scale correction. This step is useful in every document analysis task. We have focused on the structured document where to accommodate the handwritten text within a frame, words are written with variable inter-character spacing. To address this issue, a horizontal scale correction approach is proposed so that the inter-character spacing is

approximately equal. The proposed method works in two steps. In the first step the form fields called cells are extracted using the modified region growing algorithm, after which horizontal scale correction is applied to the handwritten content of each cell. This horizontal scale correction step is applied as a preprocessing step in the recognition system proposed by (Almazán et al. [2014]). The reduced error rate shows its effectiveness in the recognition task. Also, our method is compared with the horizontal normalization technique proposed in (Marti and Bunke [2001]). Qualitative analysis shows better results for our method. The proposed system has made an assumption that the location of printed and handwritten words in a form is known in advance. With the same assumption, we proceed to our next chapter where we perform association between cells of the handwritten form image.