

Handwritten and Typewritten Word and Character Separation in Unconstrained Document Images

J. Tse, D. Curtis, J. Bunch, C. Jones, E.A. Yfantis, and A. Thomas

Digital Image Processing Laboratory
Department of Computer Science
University of Nevada, Las Vegas
Las Vegas, Nevada, USA

Abstract *Separating handwritten and typewritten text within unconstrained paper documents can provide more accurate and efficient OCR results. This paper presents a technique developed that can isolate both the typewritten and handwritten portions of a document image. The classification between handwritten and typewritten text occurs at both the character and the word level. Characters are grouped into words using a word separation technique with an “island” grouping method. Structural features of handwritten and typewritten characters are examined. The method developed correctly identified with probability close to 100% of the total number of typewritten words in a set of 30 handwritten documents.*

Keywords: Document Image Analysis, OCR, word separation

1 Introduction

In performing OCR in unconstrained document images, it is important to make a distinction between handwritten and typewritten text because both require its own OCR engines. Techniques for typewritten and handwritten word classification can occur at three basic levels: the line level, the word level, and the character level. More literature was found on separation that occurs at the line level. Kavallieratou et al analyzed the horizontal profile of a text line [1]. They observed that typewritten characters on a line have a relatively stable height whereas the height of each character may vary in handwritten text lines. Classification was based on discriminant analysis. An average accuracy of 98.2 % was achieved. Fan et al focused on using a word block layout to identify typewritten text lines [2]. They observed that typewritten text lines are written straight, such that a line connecting the

center low point of each word is fairly straight. An accuracy of 86 % was achieved.

Separation has also been performed on the word level [3]. A total of 31 features were extracted, and trained Fisher classifiers were used to identify typewritten and handwritten text. A Markov Random Field approach was used to rectify misclassified words. An accuracy of 98 % was achieved. It was noted that the mean and variance of the width of each character is consistent in typewritten words. Also characters of typewritten words are less likely to overlap than handwritten words. Another method analyzed the vertical projection profile of the word. Since handwritten characters may be touching each other, peaks and valleys of the connected components will not be easily identified. A Hidden Markov Model was used to classify typewritten and handwritten words based on vertical projection profiles [4]. This method achieved a precision of 92.86 %.

Analysis on the character level is relatively more difficult since less information is available. One method analyzed physical features such as the straightness of lines, and character symmetry [5]. Individual character classification reached an accuracy of 78.5 %. The method we developed is implemented at the character level similar to [5]. In unconstrained document images, words may appear in arbitrary locations. Separation at the character level allows for classification to occur regardless of word location or arrangement. This is our motivation for analysis on the character level.

Our method assumes that all invariant features have been removed from the document image. This research is performed for document recognition and indexing applications for the Department of Energy. Our method is implemented after the document feature extraction has been performed [6]. The algorithm is implemented on an image that

contains only handwritten and typewritten text. The text is then grouped into islands and each character is analyzed.

Our method was tested on a corpus of 30 document images selected from the medical record research project for the Department of Energy. Of all the typewritten words in these documents, our system correctly identified close to 100% of the typewritten words.

This paper is organized as follows. Section 2 will be the algorithm used in island grouping. Then we will present our algorithm in section 3 and describe the three features we used. Finally, we will conclude in section 4.

2 Island Grouping

After all features have been removed as discussed in [6], the only objects contained in the image are words. The first step is to isolate lines of words that satisfies the following constraints:

- a) The line consists of letters of the same font and size
- b) The letters are linearly aligned so that all letters are on the same horizontal line

To do this, we first find the connected components of the image. Then the letter components are sorted based on the left most coordinate of each component. Finally, the characters are grouped based on two conditions:

- a) Linearly centered proximity to account for vertical orientation
- b) Left and right threshold for spans of empty space between segments

Each character that passes this test is merged into the corresponding section to which it belongs, and these groups ultimately become the islands. After the completion of this algorithm, the result is a list of islands which will then be used for word separation.

3 Typewritten word isolation

We begin classification on the character level, using the word islands obtained in section 2. Three main features of typed characters are extracted from the connected component: 1)horizontal and vertical lines 2)the existence of a loop and loop symmetry and 3)character symmetry. The existence of one of

the three above properties is sufficient to classify all the typed uppercase and lowercase characters of the English alphabet.

Poor quality images may alter the effectiveness of feature extraction. To counter this potential erroneous effect, we further apply classification on the word level based on a weighted probability. Each character of the word is classified as either handwritten or typewritten. Based on the strict nature of the rules in identifying typewritten characters, a ratio 3:5 of typewritten to handwritten characters is sufficient to classify the word as typewritten. The weights are based on experiments that show a 67% confidence in typewritten classification. So, given a count of the number of typewritten components found in an island (T) versus the count of the number of handwritten components found in an island(H), the island's classification C is

$$C = \begin{cases} t & .63(T) \geq .37(H) \\ h & .63(T) < .37(H) \end{cases} \quad (1)$$

where t is the classification for typewritten words and h is the classification for handwritten words.

3.1 Horizontal and Vertical Line Detection

We define vertical and horizontal lines as lines that stretch across the entire connected component (fig 1). By observation, it is difficult and unlikely to hand write vertical/horizontal lines that extend the entire height/width of the component. By requiring lines to extend the entire length of the component, we are strengthening our classification method. To find the lines, we find (1) the thickness of the vertical/horizontal segments of the character, and (2) the vertical/horizontal projection profiles of the character. Since the methods for determining vertical and horizontal features are similar, from this point forward, we will only focus on the vertical features.

3.1.1 Line Thickness

Vertical line thickness and horizontal line thickness may be different in typed characters, and must be dealt with separately, but using the same approach. To determine the vertical line thickness in pixels, the character image is thinned using the method described in [7]. Every pixel at coordinate $P(x,y)$ of the thinned image is scanned for the vertical line property:

$$\exists q_1 q_2 \in N_p | (q_{1y} = P_y - 1 \wedge q_{1x} = P_x) \wedge (q_{2y} = P_y + 1 \wedge q_{2x} = P_x) \quad (2)$$

where q_1 and q_2 are black pixels of the thinned image and N_p is the neighborhood set of pixels of P . If P satisfies the vertical line property, horizontal scans of the original image will originate from point P and travel outwards until a white pixel is found. The thickness is the sum in pixels of the left and right scans.

The quality of the thinner may cause arbitrary pixels to satisfy the vertical line property. However, this is both arbitrary and inconsistent, so thickness values associated with false vertical lines will deviate. True vertical lines are uniform in thickness, and stretches the entire height of the component. Every pixel of the line will return a consistent thickness value. Therefore the mode of all vertical thickness values calculated represents the correct thickness of the vertical line.

3.1.2 Projection Profile

A vertical projection profile combined with the vertical line thicknesses is used to identify the existence of vertical lines in the character (fig 1). For a line to be considered a vertical line, it must satisfy the following two properties:

- a) All pixels of the line must stretch the height of the character.
- b) The thickness of the line must be equal to the thickness found in 3.A.1.

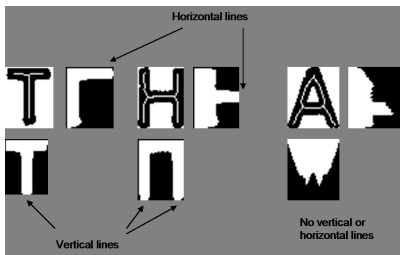


Figure 1: Horizontal and vertical line detection.

3.2 Loop Detection and Loop Symmetry Analysis

Observations indicate typewritten characters consisting of loops have loops that are highly symmetrical. Our method takes into consideration loops

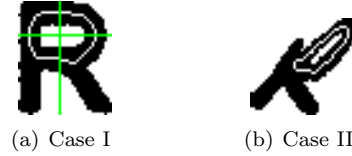


Figure 2: Vertical and Horizontal loop symmetry

P 1	8	7
2	C	N 6
3	N 4	5

Figure 3: Traversing pattern for loop detection.

that have point symmetry as well as line symmetry, such as the ‘D’ and the ‘B’. This eliminates handwritten characters that may possess solely of point symmetry (fig 2). This aspect strengthens our classification method. If an upper or lower case character (except lower case ‘a’) possess one or more loops, they are either horizontally symmetrical or vertically symmetrical, or both. Our approach is sufficient in classifying these characters. We calculate loop symmetry by finding (1) the existence of a loop and (2) vertical and horizontal line symmetry levels with respect to a vertical/horizontal line drawn at its center of mass.

3.2.1 Loop Detection

To detect the existence of a loop, we first thin the character image. The image is scanned from left to right until it twice crosses from a white to a black pixel. The two crossing points represent potential loop boundaries. To verify the existence of the loop, we pick one of the points and trace its path around the loop until it reaches itself. If a path cannot be found that begins and ends at the same starting point, the loop does not exist.

The tracing algorithm works as follows (fig 3): Assume P is the previous pixel of the trace and C is the current pixel. N represents possible next pixels along the tracing path. Starting from P , we traverse C ’s neighbors in a counter-clockwise manner until it reaches an N . N becomes the new C , C becomes P and the process repeats.

3.2.2 Loop Symmetry levels

Using the loop detected in section 3.2.1, we find its center of mass. Next we find the vertical loop

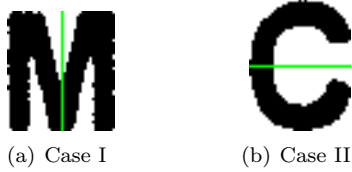


Figure 4: Vertical and Horizontal character symmetry

symmetry level with respect to a vertical line drawn at the center of mass. The same is later done for horizontal symmetry. Let $P(X_c, Y_c)$ be the center of mass of loop L , and p be a point in L .

$$\forall p \in L : p(x, y) = \begin{cases} 1 & \text{if } (2X_c - x, y) \text{ is black} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

The vertical symmetry level V_{sym} is defined as:

$$V_{sym} = \frac{1}{N} \sum_{i=0}^N p_i(x, y) \quad (4)$$

where N is the total number of black pixels in L .

3.3 Character Symmetry Analysis

Character symmetry analysis was implemented in [5]. However, like loop analysis [5] extracted point symmetries as features. We noted that only the characters ‘H’, ‘I’, ‘O’, ‘S’, and ‘X’ have point symmetry. We combine the point symmetry approach with line symmetry to identify additional characters such as ‘M’ and ‘D’. Handwritten characters are unlikely to be written perfect enough on a consistent basis to impact the identification of typewritten words.

For vertical line symmetry levels, we reflect the character image over a vertical line drawn at its center of mass. For point symmetry, we follow [5]’s method of reflecting each pixel with respect to its center of mass. Fig 4 shows some characters with vertical and horizontal symmetry.

4 Conclusion

We perform handwritten and typewritten word separation on a total of 30 document images. An average of 93.2% of the typewritten words of the image was classified correctly. It falsely identified 17.5% of the handwritten words as typewritten words. Each form has an average of a total of 123 typewritten words and 49 handwritten words.

Our classification techniques are especially vulnerable to italicized and moderately smeared words. This is precisely the drawback of this algorithm. The advantage of our approach is its simplicity. We were able to achieve competitive results by using the combination of a word and character level approach.

Our future work will be to improve handwritten and typewritten text separation by incorporating the use of grey scale document images. Grey scale images contain more data than binary images and we plan on using the extra knowledge to our advantage.

References

- [1] E. Kavallieratou and S. Stamatatos, “Discrimination of Machine-Printed from Handwritten Text Using Simple Structural Characteristics,” *17th International Conference on Pattern Recognition - Volume 1*, 2004.
- [2] K.C. Fan, L.S. Wang, and Y.T. Tu, “Classification of Machine- Printed and Handwritten Texts Using Character Block Layout Variance,” *Pattern Recognition*, vol. 31, no. 9, 1998.
- [3] Y. Zheng, H. Li, and D. Doermann. “Machine printed text and handwriting identification in noisy document images.” Technical report, LAMP Lab, UMD, College Park, 2002.
- [4] J.K. Guo and M.Y. Ma, “Separating Handwritten Material from Machine Printed Text Using Hidden Markov Models,” *Proc. Intl Conf. Document Analysis and Recognition*, 2001.
- [5] K. Kuhnke, L. Simoncini, and Z.M. Kovács-V, “A System for Machine-Written and Handwritten Character Distinction,” *Proc. Int’l Conf. Document Analysis and Recognition*, 1995.
- [6] J. Adams, E.A. Yfantis, D. Curtis and T. Pack, “Feature Extraction Methods for Form Recognition Applications,” *WSEAS trans. on Information Science and Applications*, Issue 3, Volume 3 March 2006 Pages 666-671.
- [7] M. Ahmed and R. Ward, “A Rotation Invariant Rule-Based Thinning Algorithm for Character Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v.24 n.12, p.1672-1678, 2002.