# Text Line Segmentation Based on Morphology and Histogram Projection

Rodolfo P. dos Santos, Gabriela S. Clemente, Tsang Ing Ren and George D.C. Calvalcanti
*Center of Informatics, Federal University of Pernambuco*
*Recife, PE, Brazil - www.cin.ufpe.br/~viisar*
*{rps2,gsc2,tir,gdcc}@cin.ufpe.br*

## Abstract

*Text extraction is an important phase in document recognition systems. In order to segment text from a page document it is necessary to detect all the possible manuscript text regions. In this article we propose an efficient algorithm to segment handwritten text lines. The text line algorithm uses a morphological operator to obtain the features of the images. Following, a sequence of histogram projection and recovery is proposed to obtain the line segmented region of the text. First, an Y histogram projection is performed which results in the text lines positions. To divide the lines in different regions a threshold is applied. After that, another threshold is used to eliminate false lines. These procedures, however, cause some loss on the text line area. So, a recovery method is proposed to minimize this effect. In order to detect the extreme positions of the text in the horizontal direction, an X histogram projection is applied. Then, as in the Y direction, another threshold is used to eliminate false words. Finally, in order to optimize the area of the manuscript text line, a text selection is carried out. Experimental results using the IAM-database showed that this new approach is robust, fast and produces very good score rates.*

## 1. Introduction

Text line extraction or segmentation is an important problem that does not have an universal accepted solution in the context of automatic handwritten document recognition systems [1]. Text characteristics can vary in font, size, shape, style, orientation, alignment, texture, color, contrast and background information. These variations turn the process of word detection complex and difficult [2]. In the case of handwritten manuscripts, differently from machine printed, the complexity of the problem even increases. Since handwritten text can vary greatly depending on the user skill, disposition and even cultural background.

Here, we present a method to segment text lines based on morphology and histogram projection. Morphological operations are used to produce a binary image. This procedure was first proposed by Wu et. al. [3], as an initial step in the process of text line extraction from video images containing text information. In their application, a not precise box containing the region of the text is used as output of the system to identify machine printed text in different video contexts. We have adapted and improved this idea for handwritten text line segmentation problem. An important fact in relation to image analysis based on contrast is that this characteristic is robust in relation to changes in illumination and it is invariant to different image transformations such as scaling, translation and skewing.

Once the page document has been preprocessed, a technique based on projection profiles is applied. Projection profiles are commonly used for printed document segmentation and can also be adapted to handwritten documents [1]. In the work of Marinai and Nesi [4], the projection curves are used to segment music sheets in order to extract the basic symbols and their positions. The segmentation approach proposed is divided in 3 levels and utilizes projection profiles along the Y and X axes alternately. Manmatha and Rothfeder [5] used projection profiles in the horizontal direction to segment words of historical handwritten documents during the line segmentation stage.

In this work, a projection profile in the horizontal direction is initially applied to obtain the text lines positions. Some improvements were necessary in this procedure to correctly identify the line segments, so a recovery process is also developed. A similar process is used to obtain the word borders of a line using projection profiles in the vertical direction. We refer to projection profile as histogram projection. Experiments are performed on handwritten documents randomly selected from the IAM-database [6], showing that the proposed technique produces encouraging results. An

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.183

analysis based on a performance evaluation method described in [7] is carried out.

This work is organized as follows. In Section 2, the proposed method and all its stages are described. In Section 3, the experimental results are analyzed and discussed. Finally, in Section 4, some discussion and concluding remarks are presented.

## 2. Proposed Method

In this section we propose a new method to automatically identify and segment the text line regions of a handwritten document. The system consists of eight stages, as shown in Figure 1 and described below.

The feature extraction or binarization step is applied to the input image, resulting in Figure 2b. Then, an Y histogram projection (Fig. 2c) is obtained to detect the possible lines. Due to some noises, a text line separation (Fig. 2d) is necessary. Once the false lines are found, they must be excluded. After that, the line region recovery step (Fig. 2e) is performed in order to recover some losses introduced by the preceding step. An X histogram projection that is applied to each line detected (Fig. 2f) takes out possible false words, mainly at the lateral edges of the page. Finally, we obtain the text lines region (Fig. 2g).

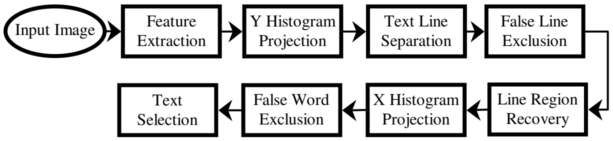


**Figure 1. Flowchart of the proposed system, showing the eight stages of the procedure.**

### 2.1. Binarization

Text in handwritten document image, most of the times, have high contrast in relation to the background of the images. The widths and heights of the text lines are relatively uniform and have a horizontal direction. The feature extraction used here is the method proposed by Wu et. al. [3], which is based on morphological operators. The original idea of this technique was to extract high contrast regions from a video image, where there is a lot of information in the background. The authors main objective was to be able to localize text region from video image in a real outdoors environment. Figure 3 shows the flowchart of this operation along with the morphological operations used to obtain the desired results.

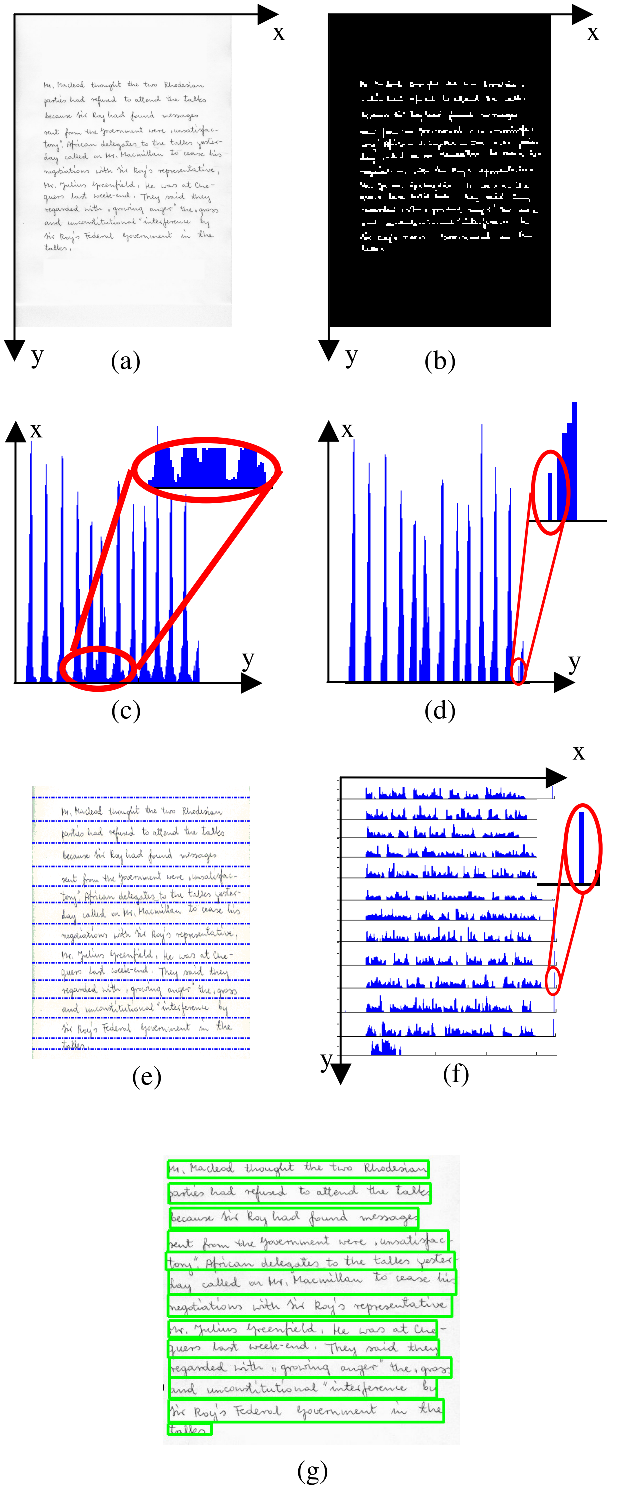
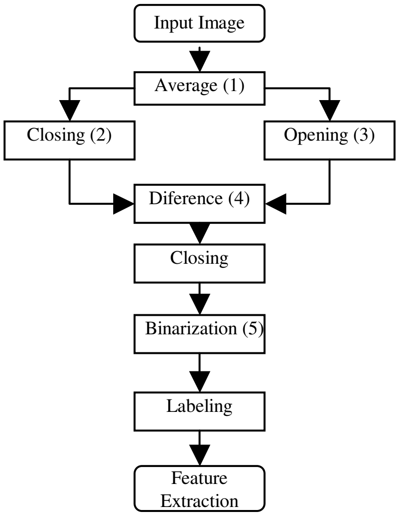


**Figure 2. Intermediate stages: (a) Input image, (b) Feature extraction step, (c) Y histogram projection with intersection between letters highlighted, (d)Text line separation with a false line highlighted, (e) Line region recovery, (f) X histogram projection with a false word highlighted and (g)Text selection.**



$$E_{S_{mxn}}(I(x,y)) = \frac{1}{mn}\sum_{i=-m/2}^{m/2}\sum_{i=-n/2}^{n/2} I(x+i,y+j)S_{m,n}(i,j) \qquad (1)$$

$$I(x,y)\bullet S_{m,n} = (I(x,y)\oplus S_{m,n})\odot S_{m,n} \qquad (2)$$

$$I(x,y)\circ S_{m,n} = (I(x,y)\odot S_{m,n})\oplus S_{m,n} \qquad (3)$$

$$D(I_1,I_2) = | I_1(x,y) - I_2(x,y) | \qquad (4)$$

$$T(I(x,y)) = \begin{cases} 255, & if \quad (I(x,y) > T; \\ 0, & otherwise \end{cases} \qquad (5)$$

**Figure 3. Flowchart of feature extraction stage.**

In equations 1-5, I(*x,y*) denotes the gray level value of the pixel located at position (*x,y*) and $S_{m,n}$ is the structural element of size m x n where m and n are odd values lager than zero. Figure 4 shows two kinds of structural elements used in mathematical morphology operations.

$$S_{1,5} = \begin{array}{|c|c|c|c|c|}\hline 1&1&1&1&1\\ \hline\end{array} \qquad S_{3,7} = \begin{array}{|c|c|c|c|c|c|c|}\hline 1&1&1&1&1&1&1\\ \hline 1&1&1&1&1&1&1\\ \hline 1&1&1&1&1&1&1\\ \hline\end{array}$$

(a) (b)

**Figure 4. Structure elements masks used by the morphological operators.**

To obtain the binarization resulting output, first a smoothing operation is used by applying an average filter that have a window size of 3 pixels, which is defined as a structure element $S_{3,3}$ in mathematical morphology context. The next operations are closing and opening using a structural element $S_{1,20}$. The difference obtained from subtracting both images is the result of the following step. Then, another closing operation is performed with a structural element $S_{5,5}$ to turn the border of the resulting image more compact and closer. Afterwards, a threshold procedure is applied followed by a labeling process to extract the text segments. In the threshold procedure a parameter T is defined dynamically according to the background of the image. This parameter is responsible to determine the limit value of the binarization operation. An example of the result of this process is shown in Figure 2b.

## 2.2. Y Histogram Projection

Once the feature extraction of the images is performed, the Y histogram projection of the whole image is obtained. The idea is to use a simple and fast method to correctly distinguish possible line segments in the handwritten text. In Figure 2c it is clear that each text line corresponds to a peak in the histogram. The histogram represents the added pixels for each y value. So the empty spaces between the peaks represent possible regions between different text lines.

## 2.3. Text Line Separation

Once all the potential lines are detected, a procedure to apply a threshold is performed to obtain a possible line separation in the text. This threshold is dynamically calculated and it is proportional to the average length of the lines in the text (Y histogram values). This process applied to the histogram aims to remove the regions in the histogram that are not referred to the lines in the text, or the elimination of noises that confuses with the text lines. The choice of the parameter to be used as threshold is intrinsic related to the information in the text, so that the algorithm utilized the minimum possible of heuristic techniques to determine the line separation points.

Actually, this stage tries to identify the location of each text line. The separation of the possible text lines regions using the histogram shows a difficult due to the upper and lower regions of some letters as shown in Figure 5.

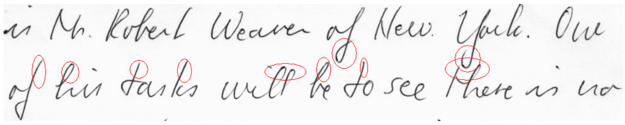

is Mr. Robert Weaver of New York. One
of his tasks will be to see there is no

**Figure 5: Regions that provoke false lines.**

The red marked areas in Figure 2c shows a detailed view of these regions in the histogram. To separate the text lines, we exclude part of the histogram based on the average Y histogram values. However, by doing so, part of the letters is lost since they are located in the region of the exclusion. To solve this problem a recovery method is also proposed.

As a result of the stage described, we obtained the new Y histogram projection, shown in Figure 2d. Note that the peaks are clearly separated by empty regions compared to Figure 2c.

### 2.4. False Line Exclusion

This procedure tries to exclude possible noises close to the text lines regions. Once the possible text line regions are separated by removing an offset from the histogram, we determine the average height of these regions to exclude false lines that might be detected. In Figure 2d we can observe this effect, a small peak in the histogram shown in red ellipse. If this region poses enough height it can be confused with a text line segment by the algorithm. The height of a line is obtained by taking the limit values of the corresponding region in the Y histogram and calculating the difference between them. The equation bellow provides the average height of the lines found in a page:

$$\sum \frac{\left| y_{initial} - y_{final} \right|}{N_p}, \qquad (6)$$

where $y_{\text{initial}}$ is the y position where the text region begins, $y_{\text{final}}$ is the y position where the text region ends and $N_p$ is the number of regions found in the page.

The lines with height below a pre-determined threshold are removed. The value of this threshold is proportional to the average height of the text lines in the whole image.

### 2.5. Line Region Recovery

This procedure determines the average point between the regions found. The idea is to find the maximum area that each line might be inscribed, by determining the superior and inferior coordinates in the y axis. Figure 2e shows the limits of these regions after the exclusion threshold is applied. The dashed lines are the limits between two adjacent line regions. In this way, the excluded regions are recovered. Note that the limit lines establish the maximum and minimum y coordinates for each text line.

### 2.6. X Histogram Projection

A histogram was also projected in the X direction for each line segmented. This procedure determines the word positions in the text line. The result of this stage is crucial to detect noise (or false words) in the image.

### 2.7. False Word Exclusion

The X projection is also used to remove region noise, in the same way that it is described in section 2.4. Figure 2f shows this effect, the red region shows a false word region that should be excluded. In this way we determine the extreme points of each sentence or the word border points (right and left). Besides that, for the precise determination of the maximum border points, four times the maximum word width among the words excluded is added. This value was obtained through several experiments.

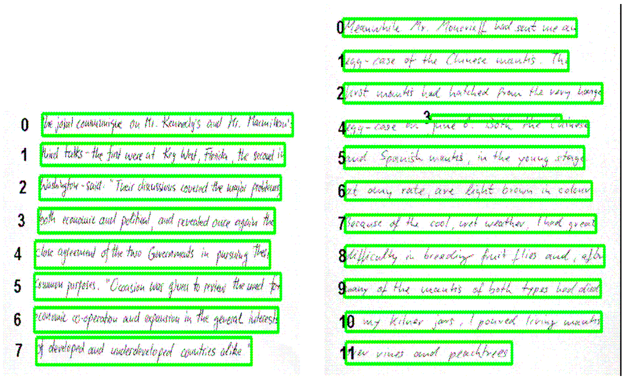
### 2.8. Text Selection

Once identified the coordinates that delimits the possible text region, an algorithm is applied to find out the smallest region "box" that inscribes that text region. The procedure is similar to the algorithm used in [3]. It consists in optimizing the dimensions of the rectangle that encompasses the text line. This region is defined as the final segmented text area and it should not cross over parts of the word or have larger dimensions than the line. Figure 2g shows an example of this procedure and the final result of the segmentation process.

## 3. Experimental Results

The experiments of the proposed method are done using 150 images (total number of text lines = 1353) randomly chosen from the IAM handwritten database. This public available database [6] provides images with 300 dpi, and 256 gray levels in a tiff graphic format. The database also provides the information about the segmented line regions, so we compared our results with the one provided by the database.

Figure 6 shows two examples of the text line segmented regions obtained by the proposed method. Note that in Figure 6a all the segments were correctly identified. However, in Figure 6b we show a case where the method still fails because line 3 is a false line. However this kind of problems occurred in less the 1,2 % of the total processed text lines.



(a) (b)

**Figure 6. Exemple of two image documents and the segmented region obtained by the algorithm.**

To evaluate the performance of the procedure we compared each segmented line position, that is the x,y of the leftmost position, the height (h) and length (w) of the box, to the ones informed by the database. We obtained agreement in pixel position level with a very small deviation. For example in Figure 6(a) lines 3 to 7 all the pixels position coincides. And for lines 0 to 2, an disagreement of less the 5 pixels was obtained for all variables x, y, w and h.

The performance measurement of the algorithm was evaluated according to the method described in [7]. Table 1 shows the detection rate and the missed detection rate, using the formula provide by equations below:

$$DetectionRate = w_1 \cdot \frac{one2one}{N} + w_2 \cdot \frac{g\_one2many}{N} + w_3 \cdot \frac{g\_many2one}{N}$$

$$MissedDetectionRate = \frac{misses}{N}$$

**Table 1. Results (Total lines = 1353)**

| Acceptance Threshold | Detection Rate | Missed Detection Rate |
|---|---|---|
| **0.50** | 0.99 | 0.01 |
| **0.55** | 0.99 | 0.01 |
| **0.60** | 0.99 | 0.01 |
| **0.65** | 0.98 | 0.02 |
| **0.70** | 0.96 | 0.04 |
| **0.75** | 0.91 | 0.09 |
| **0.80** | 0.86 | 0.14 |
| **0.85** | 0.82 | 0.18 |
| **0.90** | 0.76 | 0.24 |
| **0.95** | 0.67 | 0.33 |

The false alarm rate was obtained by:

$$FalseAlarmRate = \frac{false\_alarms}{M}$$

Using a set of ground-truth with 1353 lines in our experiments, the false alarm rate is equal to 15/1363 = 0.0111. This means that only 15 false lines are detected.

## 4. Conclusions and Discussions

In conclusion, we presented a novel algorithm for text line segmentation. The procedure proved to be fast and very accurate using the well-known IAM database. The results, even though we used a small part of all the images contained in the database, showed to be very encouraging. Text line segmentation is just the first phase for the solution of a automatic handwritten document recognition system. The next step of the problem is to segment each word of the text line and then each letter of the word. We believe that this procedure can be achieved by using the same kind of ideas used here in identifying the text line. We are currently investigating this solution.

## References

[1] L. Likforman-Sulem, A. Zahour, B. Taconet, “Text line segmentation of historical documents: a survey”, *International Journal on Document Analysis and Recognition*, 2007, pp. 123-138.

[2] K. Junga, K.I. Kimb, A.K. Jain, “Text information extraction in images and video: a survey”, *Pattern Recognition*, 2004, pp. 977-997.

[3] J.C. Wu, J.W. Hsieh, Y.S. Chen, “Morphology-based text line extraction”, *Machine Vision and Applications*, 2008, pp. 195-207.

[4] S. Marinai, P. Nesi, “Projection Based Segmentation of Musical Sheets”, *Document Analysis and Recognition,* ICDAR 1999, pp. 515-518.

[5] R. Manmatha, J.L., Rothfeder, “A scale space approach for automatically segmenting words from historical handwritten documents”, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2005, pp. 1212-1225.

[6] U.V. Marti, H. Bunke, “The IAM-database: an English sentence database for offline handwriting recognition”, *International Journal on Document Analysis and Recognition*, 2002, pp. 39-46.

[7] B. Gatos, A. Antonacopoulos, N. Stamatopoulos, “Handwriting Segmentation Contest”, *Document Analysis and Recognition*, ICDAR 2007, pp. 1284-1288.