A Block-Based Hough Transform Mapping for Text Line Detection in Handwritten Documents

G. Louloudis, B. Gatos, I. Pratikakis, K. Halatsis

A Block-Based Hough Transform Mapping for Text Line Detection in Handwritten Documents

G. Louloudis[1], B. Gatos[2], I. Pratikakis[2], K. Halatsis[1]

[1]Department of Informatics and Telecommunications,
University of Athens, Greece
http://www.di.uoa.gr
louloud@mm.di.uoa.gr,
halatsis@di.uoa.gr

[2]Computational Intelligence Laboratory,
Institute of Informatics and Telecommunications,
National Center for Scientific Research "Demokritos",
GR-153 10 Agia Paraskevi, Athens, Greece
http://www.iit.demokritos.gr/cil
{bgat,ipratika}@iit.demokritos.gr

Abstract

In this paper, we present a new text line detection method for unconstrained handwritten documents. The proposed technique is based on a strategy that consists of three distinct steps. The first step includes preprocessing for image enhancement, connected component extraction and average character height estimation. In the second step, a block-based Hough transform is used for the detection of potential text lines while a third step is used to correct possible false alarms. The performance of the proposed methodology is based on a consistent and concrete evaluation technique that relies on the comparison between the text line detection result and the corresponding ground truth annotation.

Keywords: Document Analysis, Unconstrained Handwriting, Hough Transform, Text Line Detection, Connected Component Analysis.

1. Introduction

Text line detection is a critical stage towards unconstrained handwritten document recognition. It refers to the segmentation of a document page image into distinct entities, the text lines. Representative problems that appear in this stage are the difference in the skew angle between lines on the page, overlapping words and adjacent lines touching. Also, there are cases where there are skew angle changes along the text line. Furthermore, the frequent appearance of accents in many languages (eg. French, Greek) makes the text line detection a challenging task.

In this paper, we present a new text line detection method for unconstrained handwritten documents using a block-based Hough transform. The main novelties of the proposed approach are (i) the partitioning of the connected component space into three subsets each treated in a different manner and (ii) the splitting of connected components into equally spaced blocks each of them voting in the Hough domain. This method can handle with very good accuracy documents with varying skew angle, overlapping words as well as documents with frequently appearing accents.

The paper is organized as follows: in Section 2, the related work is described. In Section 3 the method to segment text lines is detailed. Section 4 deals with the performance evaluation methodology. In Section 5 we present the experimental results and, finally, Section 6 describes conclusions and future work.

2. Related Work

A wide variety of text line detection methods for handwritten documents has been reported in the literature. In [8], a technique based on the projection profile is proposed. A histogram of the pixels' intensities at each scan line is calculated. The produced bins are smoothed and the corresponding valleys are identified. These valleys indicate the space between the lines of the text. A different approach is considered in [2] where the initial image is partitioned into vertical strips. At each vertical strip, the horizontal run histogram is calculated. This technique assumes that text appearing in a single strip is almost parallel to each other. Some other methods [11], [7], [4] make use of the Hough transform. The Hough transform is a powerful tool used in many areas of document analysis that is able to locate skewed lines of text. By starting from some points of the initial image the method extracts the lines that fit best to these points. The points considered in the Hough transform are usually either the gravity centers [7] or minima points [11] of the connected components. In a recent paper [12], a fuzzy run-length is used to segment lines. This measure is calculated for every pixel on the initial image and describes how far one can see when standing at a pixel along horizontal direction. By applying this measure, a new grayscale image is created which is binarized and the lines of text are extracted from the new image. For Arabic text, a method based on the shortest spanning tree over the main strokes is computed to locate text lines [1]. In a second stage the secondary strokes are assigned to the closest main stoke. In [10] a nearest neighbor clustering of the connected components is applied. The weighted

k-means algorithm is used for grouping. In [9] the text line extraction problem is seen by the scope of Artificial Intelligence. The aim is to cluster the connected components of the document into homogeneous sets, corresponding to the text lines of the document. To solve this problem, a search over the graph that is defined by the connected components as vertices and the distances among them as edges is applied. A recent paper [13] makes use of the Adaptive Local Connectivity Map. The input to the method is a grayscale image. In this method, a new image is calculated by summing the intensities of the neighbors in each pixel in the horizontal direction. The new image is also a grayscale image. A thresholding technique is applied in the new image and the connected components are grouped into location maps by using a grouping method.

3. Method Description

Text line detection in unconstrained handwritten document images deals with the following challenges: (i) each line that appears in the document may have an arbitrary skew angle; (ii) Accents may be cited either above or below the text line (iii) Parts of neighboring text lines may be connected and (iv) Cursive words usually consist of connected characters.

To meet the aforementioned challenges, we propose a methodology which consists of the following three steps. The first step includes preprocessing for image enhancement, connected component extraction and average character height estimation. In the second step, a block-based Hough transform is used for the detection of potential text lines while a third step is used to correct possible false alarms. These stages are described in detail in Sections 3.1-3.3.

A block diagram describing the proposed methodology is given in Figure 7.

3.1. Preprocessing

First, an adaptive binarization and image enhancement technique [5] is applied. Then, the connected components of the binary image are extracted [3] and for every connected component, the bounding box coordinates and the corresponding area are calculated. Finally, the average character height *AH* for the document image is calculated [6]. We assume that the average character height equals to the average character width *AW*.

3.2. Hough Transform Mapping

In this stage, the Hough transform takes into consideration a subset (denoted as "subset 1" in Figure 1) of the connected components of the image. This subset is chosen for the following reasons: (i) It is required to ensure that components which appear in more than one line will not vote in the Hough domain; (ii) components, such as accents, which have a small size must be rejected from this stage because they can cause a false text line detection by connecting all the accents above the core text line.

The spatial domain for "subset 1" includes all components with size identified by the following constraints:

$$0.5 * AH < H < 3 * AH \qquad (1)$$
$$1.5 * AW < W$$

where *H, W* denote the component's height and width, respectively, and *AH*, *AW* denote the average character height and the average character width, respectively.

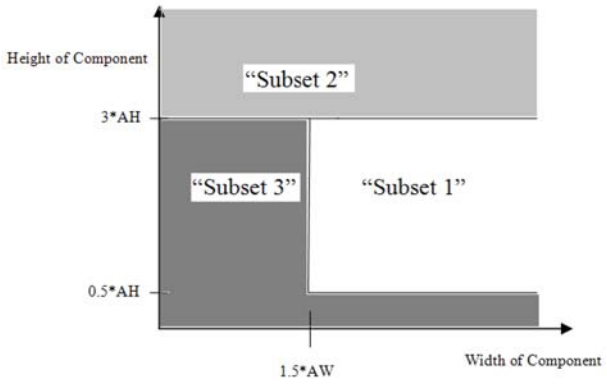

Figure 1: The connected component space partitioned to 3 subsets denoted as "Subset 1", "Subset 2" and "Subset 3".

In the classical approach [7], [4] only one representative point for a single connected component is considered to vote in the Hough domain. Instead, for each component lying in "Subset 1", a partitioning is applied so as to have more representative points voting in the Hough domain. Specifically, every connected component lying in this subset is partitioned to equally-sized blocks. The width of each block is defined by the average character width *AW*. An example is shown in Figure 2. After this partitioning stage, we calculate the gravity center of the connected component contained in each block which is then used in the voting procedure of the Hough transform.

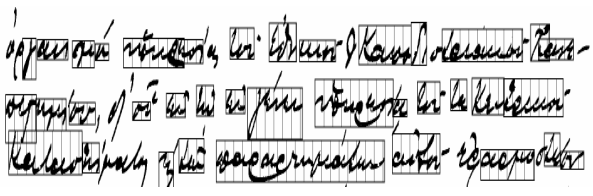

Figure 2: An example that shows the connected components partitioning into blocks of width *AW*. All connected components not placed in a bounding box correspond to either "Subset 2" or "Subset 3".

The Hough transform is a line to point transformation from the Cartesian space to the polar

coordinate space. Since a line in the Cartesian coordinate space is described by the equation:

$$x\cos(\theta) + y\sin(\theta) = p \quad (2)$$

It is easily observed that the line in the Cartesian space is represented by a point in the polar coordinate space whose coordinates are p and θ. Every point in the subset that was created above corresponds to a set of cells in the accumulator array of the (p,θ) domain. To construct the Hough domain the resolution along θ direction was set to 1 degree letting θ take values in the range 85 to 95 degrees and the resolution along p direction was set to $0.2*AH$ [4].

After the computation of the accumulator array we proceed to the following procedure: We detect the cell (ρ_i, θ_i) having the maximum contribution and we assign to the text line (ρ_i, θ_i) all points that vote in the area $(\rho_i - 5, \theta_i)..(\rho_i + 5, \theta_i)$. To decide whether a connected component belongs to a text line, at least half of the points representing the corresponding blocks must be assigned to this area. After the assignment of a connected component to a text line, we remove from the Hough transform accumulator array all votes that correspond to this particular connected component. This procedure is repeated until the cell (ρ_i, θ_i) having the maximum contribution contains less than n_1 votes in order to avoid false alarms. During the evolution of the procedure, the dominant skew angle of currently detected lines is calculated. In the case that the cell (ρ_i, θ_i) having a maximum contribution less than n_2, an additional constraint is applied upon which, a text line is valid only if the corresponding skew angle of the line deviates from the dominant skew angle less than $2°$.

3.3. Postprocessing

The previous stage may result to more than one lines assigned in the Hough domain that correspond to a single text line (see Figure 3). This correspondence is determined by calculating the distance between the corresponding crossing points of the lines with the document middle vertical line. If the distance is less than the average distance of adjacent lines then all connected components which correspond to these lines are assigned to the same text line label.

Figure 3: Two lines that need to be merged to describe one text line.

As a next step the connected components of "Subset 1" that were not clustered to a line must be checked whether they create a new line that the Hough transform did not reveal. To this end, a grouping technique of the remaining connected components is applied that utilizes the gravity centers of the corresponding blocks.

For every block with gravity center (x_i, y_i), we calculate the distance d_i between (x_i, y_i) and the closest already detected text line. If d_i ranges around the average distance of adjacent lines then the corresponding block is considered as a candidate to belong to a new text line. To decide whether a connected component is assigned to a new text line, at least half of the corresponding blocks must be candidates to belong to the new text line.

"Subset 2" includes the components whose height exceeds 3 times the average height (see Figure 1). These 'large' components may belong to more than one text line. This situation may appear when an ascender of one line meets a descender of an adjacent line. To include a connected component to a text line label, the number of lines that cross the bounding box of the connected component must be calculated. If more than one line crosses it, then this component is assigned to more than one text line otherwise it is grouped to the text line that crosses it.

"Subset 3" includes all the components that do not fall into the previous two categories. Components of "Subset 3" are usually punctuation marks or accents. As a final step all components belonging to this subset as well as the unclassified components of "Subset 1" are grouped to the closest line. In more detail, for any of these connected components, the distance from every line detected in the previous stages is calculated. This distance is the length of the vertical line that starts from the gravity center of the connected component and finishes to the point that reaches the text line.

4. Performance Evaluation Methodology

In the literature, the performance evaluation of a text line detection algorithm is mainly based on visual criteria in order to calculate the percentage of the correct segmented text lines [7], [12], [13]. Manual observation of the segmentation result is a very tedious, time consuming and not in all cases unbiased process. To avoid user interference, we propose an automatic performance evaluation technique based on comparing the text line detection result with an already annotated ground truth. Let ***DG***, ***DR*** defined as the sets of distinct regions in the ground truth and the result set accordingly:

$$\boldsymbol{DG} = \{DG_i\}, i \in [1, DGnum] \quad (3)$$

$$\boldsymbol{DR} = \{DR_i\}, i \in [1, DRnum] \quad (4)$$

and *LG, LR* defined as the text lines in the ground truth and the result set accordingly:

$$\boldsymbol{LG} = \{LG_i\}, i \in [1, LGnum] \quad (5)$$

$$\boldsymbol{LR} = \{LR_i\}, i \in [1, LRnum] \quad (6)$$

We also define functions *DLG()* and *DLR()* that correlate distinct regions with text lines in the ground truth and the result set accordingly, as follows:

$$DLG(i,j)=\begin{cases}1, \text{if } DG_i \in LG_j \\ 0, \text{otherwise}\end{cases} \quad (7)$$

$$DLR(i,j)=\begin{cases}1, \text{if } DR_i \in LR_j \\ 0, \text{otherwise}\end{cases} \quad (8)$$

For every text line LG_k, and LR_k, we define ELG_k and ELR_k as the set of regions that belong exclusively to the text line LG_k and LR_k in the ground truth and the result set accordingly, as follows:

$$\boldsymbol{ELG_k}=\{DG_i\}, i:\sum_{\text{j}} DLG(i,j)=1 \wedge DLG(i,k)=1 \quad (9)$$

$$\boldsymbol{ELR_k}=\{DR_i\}, i:\sum_{\text{j}} DLR(i,j)=1 \wedge DLR(i,k)=1 \quad (10)$$

We define that the ground truth text line LG_k has been correctly detected in the result set according to function $COR()$ which is given as follows:

$$COR(LG_k)=\begin{cases}1, \text{if } \exists k' : \dfrac{\text{Area}(\boldsymbol{ELG_k} \cap \boldsymbol{ELR_{k'}})}{\text{Area}(\boldsymbol{ELG_k})} > th \\ \qquad\qquad \dfrac{\text{Area}(\boldsymbol{ELG_k} \cap \boldsymbol{ELR_{k'}})}{\text{Area}(\boldsymbol{ELR_{k'}})} > th \\ 0, \text{else}\end{cases} \quad (11)$$

According to function $COR()$, a ground truth text line LG_k has been correctly detected only if a text line $LR_{k'}$ exists in the result set and the overlapping area of those two lines is comparable to the area of both two lines. The threshold th used for this comparison is empirically set to 0.95.

Finally, the text line detection accuracy is given by the following formula:

$$LineDetAcc=\frac{\sum_{k=1}^{LG_{num}} COR(LG_k)}{LG_{num}} 100\,\% \quad (12)$$

5. Experimental results

The proposed text line detection method is tested on unconstrained handwritten Greek documents. Very little work has been done in the field of Greek unconstrained handwritten text line detection.

We experimented with the proposed methodology using 20 document images taken from the historical archives of the University of Athens for which we have manually created the corresponding text line detection ground truth. The total number of text lines appearing on those images was 450. Parameters n_1 and n_2 in our methodology were experimentally defined to 5 and 9, respectively. Using the evaluation methodology detailed in Section 4 we achieved a line detection accuracy of 96.87%.

The resulting line detection can be shown in Figures 4 and 6. It is easily observed that there are many difficulties concerning the extraction of text lines such as the variety of accents appearing either above or under the body of the text line or even the small difference in the skew angle among the lines of text.

Most of the errors made by this method have to do with the accents. Such an example is shown in Figure 5 where the gravity center of the connected component which corresponds to an accent, is closer to the upper line rather than the lower one.

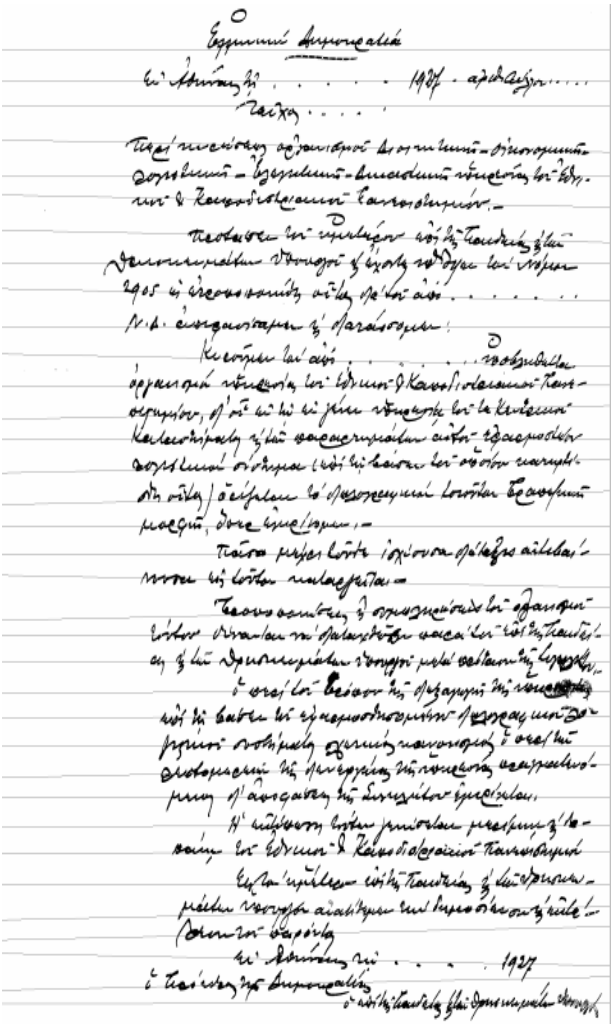

Figure 4: An image example showing the final lines created from the proposed method.

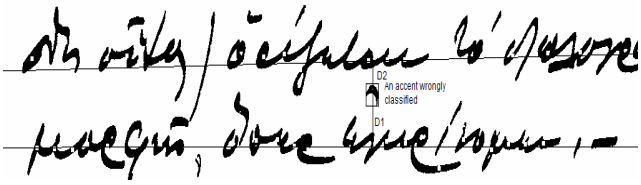

Figure 5: The accent with the bounding box is wrongly classified since D2<D1.

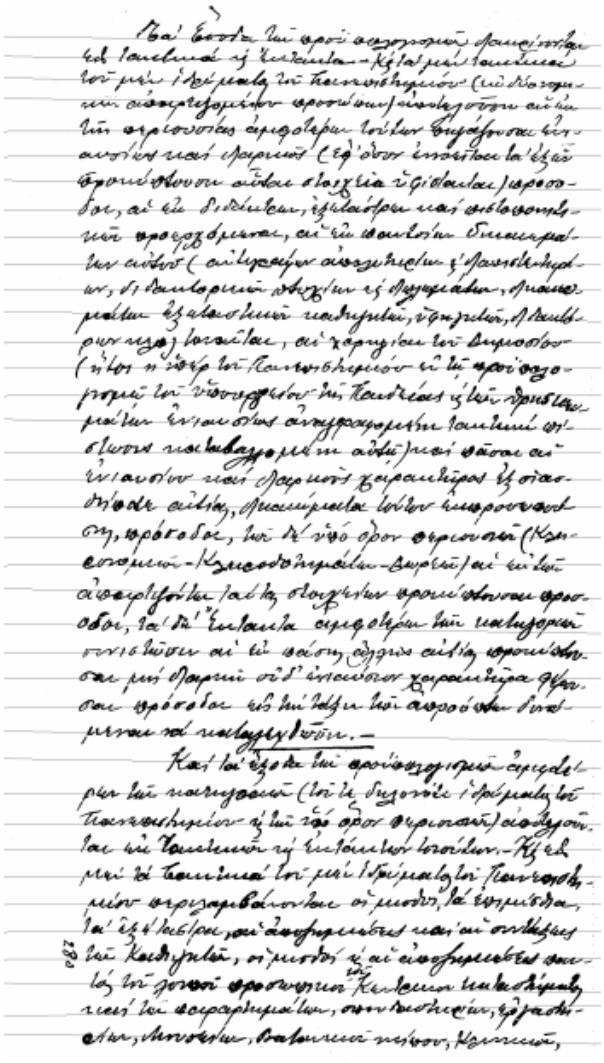

Figure 6: A second example describing the proposed method.

6. Conclusions and future work

In this paper we present a new text line detection method for unconstrained handwritten documents. The main novelties of the proposed approach consist of (i) the partitioning of the connected component space into three subsets each treated in a different manner and (ii) the splitting of connected components into equally spaced blocks each of them voting in the Hough domain. The proposed method is a sufficiently accurate method to extract the text lines from unconstrained handwritten text documents.

Future work concerns the implementation of a method that handles the difficult cases of intersection between ascending and descending strokes of adjacent lines. Another issue to handle is to find ways that correctly classify accents as they appear to cause most of the errors.

The proposed algorithm is foreseen to be implemented in a keyword spotting system specially designed for the Historical Archives of the University of Athens, Greece, where both cursive and handwritten lines of text appear.

References

[1] I.S.I. Abuhaiba, S. Datta, M.J.J. Holt, "Line Extraction and Stroke Ordering of Text Pages", *Proceedings of the Third International Conference on Document Analysis and Recognition,* Montreal, Canada, 1995, pp. 390-393.

[2] Elisabetta Bruzzone, Meri Cristina Coffetti, "An Algorithm for Extracting Cursive Text Lines", *Proceedings of the Fifth International Conference on Document Analysis and Recognition,* Bangalore, India, 1999, pp. 749.

[3] Fu Chang, Chun-Jen Chen, Chi-Jen Lu, "A Linear-Time Component-Labeling Algorithm Using Contour Tracing Technique", *Computer Vision and Image Understanding, Vol. 93, No.2,* February 2004, pp. 206-220.

[4] L. A. Fletcher, R. Kasturi, "A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images", *IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.10, No.6,* November 1988, pp. 910-918.

[5] B. Gatos, I. Pratikakis and S. J. Perantonis, "Adaptive Degraded Document Image Binarization", *Pattern Recognition, Vol. 39*, pp. 317-327, 2006.

[6] B. Gatos, T. Konidaris, K. Ntzios, I. Pratikakis and S. J. Perantonis, "A Segmentation-free Approach for Keyword Search in Historical Typewritten Documents", *8th International Conference on Document Analysis and Recognition (ICDAR'05)*, Seoul, Korea, August 2005.

[7] Laurence Likforman-Sulem, Anahid Hanimyan, Claudie Faure, "A Hough Based Algorithm for Extracting Text Lines in Handwritten Documents", *Proceedings of the Third International Conference on Document Analysis and Recognition,* Montreal, Canada, 1995, pp. 774-777.

[8] R. Manmatha, J. L. Rothfeder, "A Scale Space Approach for Automatically Segmenting Words from Historical Handwritten Documents", *IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.27, No.8,* August 2005, pp. 1212-1225.

[9] S. Nicolas, T. Paquet, L. Heutte, "Text Line Segmentation in Handwritten Document Using a Production System", *Proceedings of the 9th IWFHR,* Tokyo, Japan, 2004, pp. 245-250.

[10] L. O'Gorman, "The Document Spectrum for Page Layout Analysis", *IEEE Trans. Pattern Analysis and Machine Intelligence, vol.15, No. 11,* 1993, pp. 1162–1173.

[11] Y. Pu and Z. Shi, " A Natural Learning Algorithm Based on Hough Transform for Text Lines Extraction in Handwritten Documents", *Proceedings of the 6 International Workshop on Frontiers in Handwriting Recognition*, Taejon, Korea,1998, pp. 637-646.

[12] Z. Shi, V. Govindaraju, "Line Separation for Complex Document Images Using Fuzzy Runlength", *First International Workshop on Document Image Analysis for Libraries,* 2004, pp. 306.

[13] Z. Shi, S. Setlur, and V. Govindaraju, "Text Extraction from Gray Scale Historical Document Images Using Adaptive Local Connectivity Map", *Eighth International Conference on Document Analysis and Recognition,* Seoul, Korea, 2005, pp. 794-798.

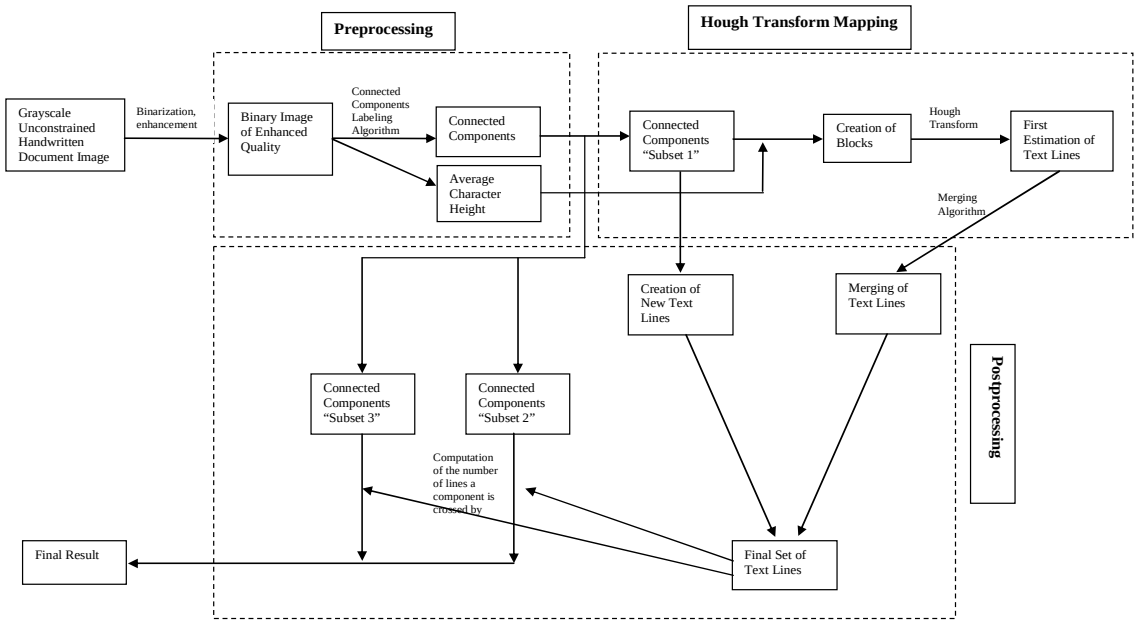

Figure 7: The block diagram of the proposed method.