A Method for Combining Complementary Techniques for Document Image Segmentation

N. Stamatopoulos, B. Gatos and S.J. Perantonis

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

Abstract

Image segmentation is a major task of handwritten document processing. Many of the proposed techniques for image segmentation are complementary, in the sense that each of them using a different approach, can solve different difficult problems such as overlapping, touching components, influence of author style etc. In this paper a combination method of different segmentation techniques is presented. Our goal is to exploit the segmentation results of complementary techniques and specific features of the initial image so as to generate improved segmentation results. Experimental results on handwriting line segmentation methods demonstrate the effectiveness of the proposed combination method.

Keywords: segmentation, handwriting, combination method, image processing.

1. Introduction

Combining classifiers is a well researched topic in the pattern recognition community [1] [2] [3]. In classifier combination, rules are used to combine the outputs of multiple classifiers. The general objective is to exploit the complementary information between the classifiers and find the rules for building hybrid classifiers that outperform their constituent classifiers. In a sense, the different classifiers in a classifier combination can be seen as a collection of weak classifiers, where each classifier can solve some different difficult problems. Some of the most common classifiers combinations methods used in the literature include voting, and liner and logistic regression.

With a similar way of thought we could combine the results of different segmentation techniques in order to achieve better segmentation results. Document segmentation into lines, words and characters is a major task in a document image analysis system [4]. There are many approaches in the literature for document segmentation which can be categorized in five major categories: projection profiles, smearing, grouping, Hough-based and stochastic methods. Techniques from each category can solve some specific problems such as overlapping, touching components, image degradations and others. Projection-based methods [5] are commonly used for printed document segmentation but can also be adapted to handwritten documents with little character overlap and they aren't sensitive to writing fragmentation. Smearing methods, such as Run-Length Smoothing Algorithm [6], enriched by local considerations can solve specific problems including overlapping, touching strokes etc. Grouping methods [7] consist in building alignments by aggregating units in a bottom-up strategy and methods based on the Hough transform [8] can be applied to fluctuating lines of handwritten drafts. Finally, stochastic methods [9] are more robust, but their implementation requires great care, particularly regarding the initialization phase.

In this paper, we propose a combination method of segmentation techniques. Our goal is to increase the efficiency and the accuracy of the segmentation result using (i) the results of segmentation techniques, which belong to different categories, as mentioned before, and (ii) specific features of the initial document according to the segmentation problem which we have to face, such as line, word or character segmentation. The rest of the paper organized as follows: In section 2 the proposed combination method for different segmentation techniques is detailed. In section 3 we present experimental results using two complementary handwriting line segmentation techniques which indicate the effectiveness of the proposed method. Finally, conclusions are drawn in Section 4.

2. Combination Method

Consider a binary image:

$$I(x,y) = \begin{cases} 1 \\ 0 \end{cases}, 1 \le x \le I_x, 1 \le y \le I_y \qquad (1)$$

where 1 and 0 correspond to the foreground and the background respectively.

Let $R_1(x, y)$, $R_2(x, y),..., R_N(x, y)$ represent the results of N different segmentation methods, which have been applied to the image $I(x, y)$, and are defined as follows:

$$R_j(x, y) \in A_j, \text{ where } A_j = \{1,...,n_j\} \quad (2)$$

Each value i in the set A_j denotes that a pixel belongs to the ith segment according to the jth method of segmentation. In figure 1 we see a specific example with $N = 3$ segmentation methods.

Our goal is to generate a new segmentation result $R(x, y)$:

$$R(x, y) \in A, \text{ where } A = \{1,...,n\} \quad (3)$$

using the segmentation results $R_1(x, y),..., R_N(x, y)$. For this reason, we define the following binary images:

$$C(x, y)_{\{r_1, r_2,..., r_N\}} = \begin{cases} 1 \text{ if } (R_1(x, y) = r_1 \text{ AND } .... \\ \quad ... \text{ AND } R_N(x, y) = r_N) \\ 0 \qquad\qquad otherwise \end{cases} \quad (4)$$

where $1 \leq r_j \leq n_j, j = 1,..., N$, which represents the intersection of the segmentation results with segment ids $\{r_1, r_2,..., r_N\}$ (see Figure 1e), and

$$D_j(x, y)_{\{i\}} = \begin{cases} 1 \text{ if } R_j(x, y) = i \\ 0 \quad otherwise \end{cases}, where \ j = 1,..., N \quad (5)$$

which represents the intersection of the image $I(x, y)$ with the ith segment according to the jth method of segmentation (see Figure 1f).

In order to represent the overlap between the segmentation results we define the following function:

$$f_j(r_1, r_2,...., r_N) = \begin{cases} \dfrac{\sum\limits_{x,y} C(x, y)_{\{r_1, r_2,..., r_N\}}}{\sum\limits_{x,y} D_j(x, y)_{\{r_j\}}} \text{ if } \sum\limits_{x,y} D_j(x, y)_{\{r_j\}} \neq 0 \\ 0 \qquad\qquad\qquad othewise \end{cases} \quad (6)$$

where $j = 1,..., N$.

For example, in Figure 1,

$$f_1(3,5,4) = \frac{\sum\limits_{x,y} C(x, y)_{\{3,5,4\}}}{\sum\limits_{x,y} D_1(x, y)_{\{3\}}} = \frac{9}{11} = 0.81$$

which means that 81% of the pixels with value 3 in result $R_1(x, y)$ have also value 5 in result $R_2(x, y)$ and value 4 in result $R_3(x, y)$.

Finally, we define the following function:

$$V = FeatureExtraction \ [Q(x,y)] \quad (7)$$

which receives as input a binary image $Q(x,y)$ and returns a vector $V = \{v_1, v_2,..., v_p\}$, where $0 \leq v_i \leq 1$, $i = 1,..., p$ representing specific features of the image $Q(x,y)$. We choose the features according to the segmentation problem which we have to face, such as line segmentation, word segmentation etc. In the Section 3 we describe specific features in order to combine line segmentation methods.
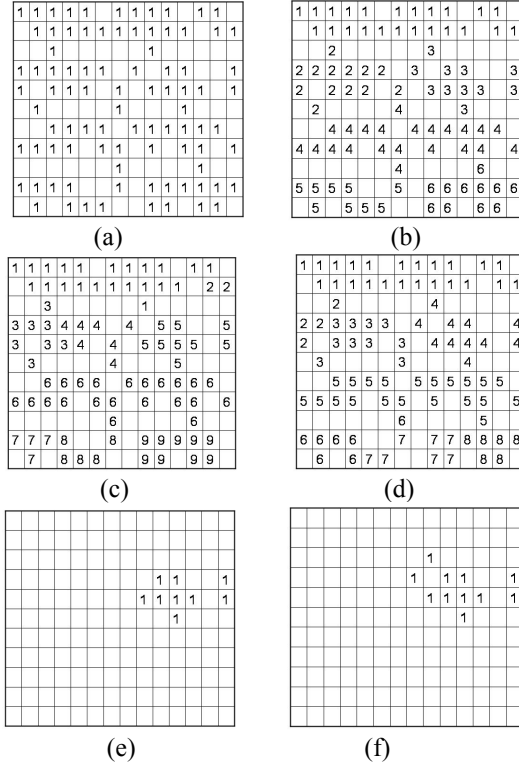
Figure 1. An example with three different segmentation results, where blank pixels represent the pixels of background (a) Binary image $I(x, y)$, segmentation results (b) $R_1(x, y)$, (c) $R_2(x, y)$, (d) $R_3(x, y)$, (e) binary image $C(x, y)_{\{3,5,4\}}$, (f) binary image $D_1(x, y)_{\{3\}}$.

Our goal is to find the segments where all the segmentation methods are in agreement and then, for the remaining segments, we use the features properly in order to decide which segments will be merged or splitted. The distinct steps we follow in order to generate the new segmentation result $R(x, y)$ are as follows:

STEP 1: At this step we detect the regions in which all the segmentation results have high degree of overlap ($\geq 70\%$). Then, we extract the features of each region and, finally, we calculate the average value of these features $AV = \{av_1, av_2,..., av_p\}$. In this way, it is hoped that the vector AV approximates the average features of the

correctly segmented regions. In order to achieve that we use the following algorithm:

CalculateVector AV [ ]

{

$\forall \{r_1, r_2, ..., r_N\} : [f_1(r_1, r_2, ..., r_N) \geq 0.7 \ AND \ ....$

$.... \ AND \ f_N(r_1, r_2, ..., r_N) \geq 0.7 \ ]$

$V_i = FeatureExtraction[C(x,y)_{\{r_1, r_2, ..., r_N\}}]$

$$AV = \frac{\sum_{i=0}^{m} V_i}{m}$$

}

where

$$m = |\{r_1, r_2, ..., r_N\}| \ where$$

$$\{r_1, r_2, ..., r_N\} : [f_1(r_1, r_2, ..., r_N) \geq 0.7 \ AND \ .... \quad (8)$$

$$... \ AND \ f_N(r_1, r_2, ..., r_N) \geq 0.7 \ ]$$

and $|\cdot|$ is the cardinality of a set.

STEP 2: At this step we detect the regions in which all the segmentation results have very high degree of overlap ($\geq 90\%$), so we decide that segmentation regions have been detected correctly, so we add them to the new segmentation result $R(x,y)$. In order to achieve that we use the following algorithm:

FindCorrectSegmentationRegion [ ]

{

$k = 0$

$\forall \{r_1, r_2, ..., r_N\} : [f_1(r_1, r_2, ..., r_N) \geq 0.9 \ AND \ ...$

$... \ AND \ f_N(r_1, r_2, ..., r_N) \geq 0.9 \ ]$

$\{ \quad R(x,y) = k * C(x,y)_{\{r_1, r_2, ..., r_N\}}$

$\quad k = k + 1 \qquad \qquad \qquad \}$

}

where $k = 1, ..., z$ and

$$z = |\{r_1, r_2, ..., r_N\}| \ where$$

$$\{r_1, r_2, ..., r_N\} : [f_1(r_1, r_2, ..., r_N) \geq 0.9 \ AND... \quad (9)$$

$$.... \ AND \ f_N(r_1, r_2, ..., r_N) \geq 0.9 \ ]$$

Looking at the example of Figure 1, we can detect results $\{1,1,1\}$ and $\{4,6,5\}$ which satisfy the above criterion:

$$f_1(1,1,1) = \frac{21}{23} = 0.91, f_2(1,1,1) = \frac{21}{22} = 0.95, f_3(1,1,1) = \frac{21}{23} = 0.91$$

and

$$f_1(4,6,5) = \frac{20}{22} = 0.9, f_2(4,6,5) = \frac{20}{22} = 0.9, f_3(4,6,5) = \frac{20}{21} = 0.95$$

Figure 2 depicts the new segmentation result $R(x,y)$ after applying Step 2 to the example of Figure 1.

Figure 2. New result $R(x,y)$ after applying Step 2 to the example of Figure 1.

STEP 3: At this step the combinations $\{r_1, r_2, ..., r_N\}$, which have been used in the previous step or they have at least one common value $r_1, r_2, ..., r_N$ with them, don't participate. In our example (see Figure 1), the combinations which don't participate are the following: $\{1,1,1\}$ and $\{4,6,5\}$ because they have been used in Step 2 and $\{1,2,1\}$, $\{3,1,4\}$, $\{4,4,3\}$, $\{4,6,6\}$ and $\{6,6,5\}$ since they have at least one common value with the combinations which have been used.

We divide the remaining combinations $\{r_1, r_2, ..., r_N\}$, where $\sum_{x,y} C(x,y)_{\{r_1, r_2, ..., r_N\}} \neq 0$, into groups $G_i, i = 1, ..., l$. Each group denotes a region of the initial image and each combination $\{r_1, r_2, ..., r_N\}$ denotes a subregion of it. Our goal is to examine, at the next step, each group separately and decide which subregions of it will be merged or splitted in order to create correctly segmented regions. In order to create the groups we use the following algorithm:

CreateGroups []

{

$i = 0$

$Until \ all \ \{r_1, r_2, ..., r_N\} : \sum_{x,y} C(x,y)_{\{r_1, r_2, ..., r_N\}} \neq 0 \ have \ joined$

$\qquad \qquad \qquad \qquad \qquad \qquad \qquad a \ group$

{

$\quad i = i + 1$

$\quad G_i = \{\varnothing\}$

$\quad Find \ \{r_1, r_2, ..., r_N\} : [\ \sum_{x,y} C(x,y)_{\{r_1, r_2, ..., r_N\}} \neq 0$

$\qquad \qquad AND \ \{r_1, r_2, ..., r_N\} \notin G_j, j = 1, ..., i-1 \ ]$

$\quad JoinGroup[\ G_i, \{r_1, r_2, ..., r_N\} \ ]$

}

}

where function $JoinGroup[\ G_i, \{r_1, r_2, ..., r_N\} \ ]$ is a recursive function and it is defined as follows:

$JoinGroup\ [\ G_i, \{r_1, r_2, ..., r_N\}]$

$\{$

   $G_i = G_i \cup \{r_1, r_2, ..., r_N\}$

   $\forall \{r_1', r_2', ..., r_N'\} : [\ \sum_{x,y} C(x,y)_{\{r_1', r_2', ..., r_N'\}} \neq 0$

            $AND\ \{r_1', r_2', ..., r_N'\} \notin G_j, j = 1, ..., i\ ]$

    $if\ (r_1 = r_1'\ OR\ ....\ OR\ r_N = r_N')$

      $JoinGroup[\ G_i, \{r_1', r_2', .., r_N'\}\ ]$

$\}$

    In our example (see Figure 1 and 2), two groups will be created:

      $G_1 = \{\{2,3,2\}, \{2,3,3\}, \{2,4,3\}, \{3,4,4\}, \{3,5,4\}\}$

and

      $G_2 = \{\{5,7,6\}, \{5,8,6\}, \{5,8,7\}, \{6,9,7\}, \{6,9,8\}\}$

Figure 3 depicts the regions of these groups and their subregions which are denoted from each combination.
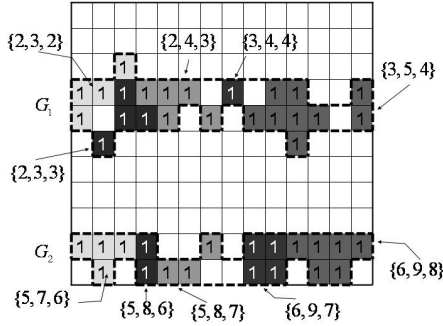
Figure 3. Regions and subregions of groups G_1 and G_2.

STEP 4: At this step we process every group $G_i, i = 1, ..., l$ independently. We start from a small subregion, in which the segmentation results have the highest degree of overlap, and then we merge other subregions with it until the features of the new region are closer to the average features AV of the image. In order to achieve that we use the following algorithm:

$ProcessGroup\ [G_i]$

$\{$

   $\forall \{r_1, r_2, ..., r_N\} \in G_i \quad g(\{r_1, r_2, ..., r_N\}) = 0$

   $k = z\ \ //see\ Eq.\ 9$

   $STOP = false$

   $Until\ (STOP = TRUE)$

   $\{$

     $k = k + 1$

// Find the subregion which has the highest overlap

$MI = \max_{\substack{\{r_1, r_2, ..., r_N\} \in G_i \\ g(\{r_1, r_2, ..., r_N\}) = 0}} \{f_1(r_1, r_2, ..., r_N), ..., f_N(r_1, r_2, ..., r_N)\}$

$Find\ \{r_1, r_2, ..., r_N\} \in G_i : [\ g(\{r_1, r_2, ..., r_N\}) = 0\ AND$

          $(f_1(r_1, r_2, ..., r_N) = MI\ OR\ .....$

           $... OR\ f_N(r_1, r_2, ..., r_N) = MI\ )\ ]$

 $If\ (\ MI < 0.2)$

   $STOP = TRUE$

 $Else$

 $\{$

    // Add this subregion to the new result

    $g(\{r_1, r_2, ..., r_N\}) = 1$

    $R(x,y) = k * C(x,y)_{\{r_1, r_2, ..., r_N\}}$

    // Calculate the difference between the featuers of

    // it and the average features of image

    $V_1 = FeatureExtraction[C(x,y)_{\{r_1, r_2, ..., r_N\}}]$

    $Dif_1 = |AV - V_1|$

    $M(x,y) = C(x,y)_{\{r_1, r_2, ..., r_N\}}$

    // Examine if we can merge other subregions

    //according to this region

    $If\ (f_j(r_1, r_2, ..., r_N) = MI\ )$

      $Merge_h[G_i, Dif_1, \{r_1, r_2, ..., r_N\}, M(x,y), k]$

          $where\ h = 1, ..., N\ AND\ h \neq j$

   $\}$

 $\}$

$\}$

where

$$g(\{r_1, r_2, ..., r_N\}) = \begin{cases} 1, & if\ \{r_1, r_2, ..., r_N\}\ has\ been \\ & added\ to\ the\ new\ result \\ 0 & otherwise \end{cases} \quad (10)$$

and function $Merge_j[G_i, Dif, \{r_1, r_2, ..., r_N\}, M(x,y), k]$, where $j = 1, 2, ..., N$ is a recursive function and it is defined as follows:

$Merge_j[G_i, Dif, \{r_1, r_2, ..., r_N\}, M(x,y), k]$

$\{$

   N(x,y)=M(x,y)

   // Add subregions with common value r_j

   $\forall \{r_1', r_2', ..., r_N'\} \in G_i : [\ r_j = r_j'\ AND\ g(\{r_1', r_2', ..., r_N'\}) = 0$

      $N(x,y) = N(x,y) + C(x,y)_{\{r_1', r_2', ..., r_N'\}}\ ]$

   // Calculate the difference between the featuers of

   // the new region and the average features of image

   $V_2 = FeatureExtraction[C(x,y)_{\{r_1, r_2, ..., r_N\}}]$

   $Dif_2 = |AV - V_2|$

If $(\mathrm{Dif}_2 < Dif\,)$

{

    // Add all the subregions to the new result

    $\forall \{r_1^{'}, r_2^{'}, ..., r_N^{'}\} \in G_i : \mathrm{r}_j = r_j^{'}$ AND $g(\{r_1^{'}, r_2^{'}, ..., r_N^{'}\}) = 0$

    {

        $g(\{r_1^{'}, r_2^{'}, ..., r_N^{'}\}) = 1$

        $R(x, y) = k * C(x, y)_{\{r_1^{'}, r_2^{'}, ..., r_N^{'}\}}$

    }

    M(x,y)=N(x,y)

    // Examine if we can merge other subregions

    //according to the new subregions

    $\forall \{r_1^{'}, r_2^{'}, ..., r_N^{'}\} \in G_i : r_j = r_j^{'}$

      $Merge_h[G_i, Dif_2, \{r_1^{'}, r_2^{'}, ..., r_N^{'}\}, M(x, y), k]$

          $where \ h = 1, ..., N \ AND \ h \neq j$

  }

}

Figure 4 depicts the distinct steps we follow in order to process $G_1 = \{\{2,3,2\}, \{2,3,3\}, \{2,4,3\}, \{3,4,4\}, \{3,5,4\}\}$ in our example, (see Figure 1, 2 and 3):

- Start from subregion $\{2,3,2\}$ ($f_3(2,3,2) = \frac{4}{4} = 1$), add it to the new result $R(x, y)$ (see Figure 4a,b) and calculate the difference between the features of it and average features AV (let $Dif_1 = 0.5$).

- Examine if we can merge with it the subregions $\{2,3,3\}$ and $\{2,4,3\}$ (because $r_1 = 2$, as $\{2,3,2\}$) (see Figure 4c), so we calculate the difference between the features of the new region and average features AV (let $Dif_2 = 0.3$).

- $Dif_2 < Dif_1$, so we add them at the new result $R(x, y)$ (see Figure 4d).

- Similarly, we examine if we can merge with it the subregion $\{3,4,4\}$ (because $r_2 = 4$ as $\{2,4,3\}$), calculate the difference (Dif_3) between the features of the new region and average features AV. (let $Dif_3 = 0.6$).

- $Dif_3 > Dif_2$, so we don't add this subregion to the new result $R(x, y)$.

With a similar process, we start from the subregion $\{3,5,4\}$, then we add the subregion $\{3,4,4\}$ (see Figure 4e), so we have the final result $R(x, y)$ after the process of group G_1 (see Figure 4f).
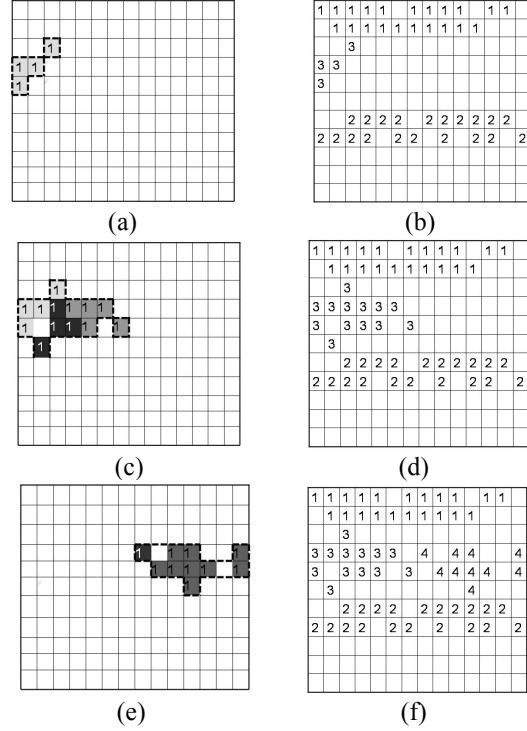

(a)   (b)

(c)   (d)

(e)   (f)

Figure 4. Apply Step 4 at group G_1 (a) $C(x,y)_{\{2,3,2\}}$ (b) new result with $C(x,y)_{\{2,3,2\}}$, (c) $C(x,y)_{\{2,3,2\}} \cup C(x,y)_{\{2,3,3\}} \cup C(x,y)_{\{2,4,3\}}$, (d) new result with $C(x,y)_{\{2,3,2\}} \cup C(x,y)_{\{2,3,3\}} C(x,y)_{\{2,4,3\}}$ (e) $C(x,y)_{\{3,5,4\}} \cup C(x,y)_{\{3,4,4\}}$ (f) new result after applying Step 4 at G_1.

STEP 5: At this step every pixel of the foreground, which doesn't have a value at the final image $R(x, y)$, inherit the value of the nearest pixel which has a value.

3. Experimental Results

To verify the validity of the proposed method we use two complementary line segmentation methods, projection profiles based on [5] and ARLSA based on [6].

We apply each method to a set of 20 handwritten documents, which are written in several languages. Then, using the two different segmentation results for each image, we generate a new segmentation result according to the proposed technique. For this reason we use the following features (see Eq. 7):

- Length of the bounding box
- Height of the bounding box
- Height to length ration of the bounding box
- Density of foreground pixels

We manually marked the correct line segments in the set of 20 images. The performance evaluation was based on counting the number of matches between the lines

detected by the individual algorithms and the lines in the ground truth [10]. The performance was recorded in terms of detection rate *(DR)* and recognition accuracy *(RA)*, while as an overall measure we used the F-measure *(FM)*:

$$FM = \frac{2DR\,RA}{DR+RA} \qquad (11)$$

As depicted in Table 1, the new segmentation result outperforms the two others methods and it increases the overall evaluation measure about 21%. Figure 5 depicts an example of the proposed combination method.

Table 1. Comparative results

	DR	RA	FM
Projection Profiles	75,1%	69,1%	71,9%
ARLSA	74,8%	71,1%	72,9%
After Combination of Projection Profiles and ARLSA	94,4%	93,7%	94,0%

(a)

(b)

(c)
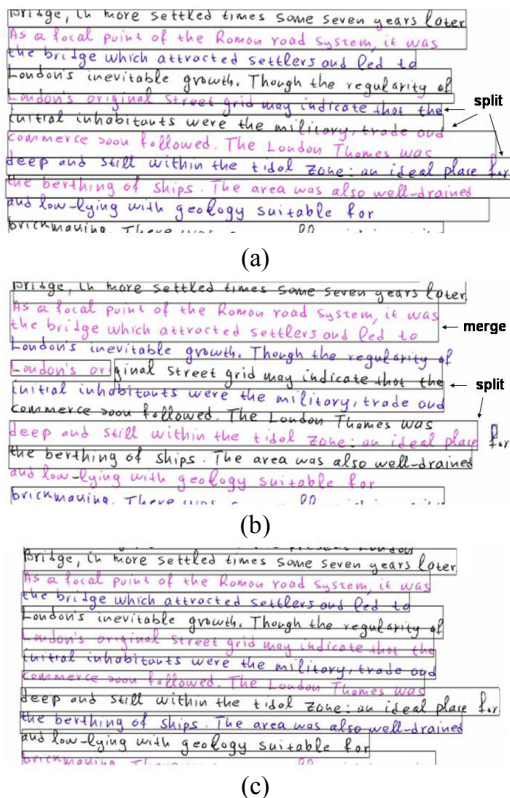
Figure 5. Segmentation results with the bounding boxes: (a) Projection Profiles (b) ARLSA (c) Final result after combination.

4. Conclusion

This paper proposes an efficient combination method of segmentations techniques. The proposed method combines the segmentation results of different techniques in order to increase the efficiency and the accuracy of the segmentation result. Also, it uses specific features of the initial document depending on the segmentation problem which we have to solve, such as line, word or character segmentation. Our future research will focus on the application of the proposed method using more segmentation techniques and in different segmentation problems, such as word and character segmentation.

References

[1] J. Kittler, M. Hatef, R. Duin, and J. Matas, *"On combining classifiers"*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, 1998, pp. 226–239.

[2] S. Günter and H. Bunke, *"Combination of three classifiers with different architectures for handwritten word recognition"*, Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition, Tokyo, Japan, 2004, pp. 63-68.

[3] M. Magimai-Doss, D. Hakkani-Tur, O. Cetin, E. Shriberg, J. Fung, and N. Mirghafori, *"Entropy Based Classifier Combination for Sentence Segmentation"*, IEEE International Conference on Acoustics, Speech and Signal Processing, Honolulu, Hawaii, 2007, pp. 189-192.

[4] L. Likforman-Sulem, A. Zahour, B. Taconet, *"Text line segmentation of historical documents: a survey"*, International Journal on Document Analysis and Recognition, No. 2-4, April 2007, pp. 123-138.

[5] A. Antonacopoulos, D. Karatzas, *"Document Image analysis for World War II personal records"*, First International Workshop on Document Image Analysis for Libraries, Palo Alto, 2004, pp. 336-341.

[6] M. Makridis, N. Nikolaou and B. Gatos, "*An Efficient Word Segmentation Technique for Historical and Degraded Machine-Printed Documents"*, 9th International Conference on Document Analysis and Recognition, Curitiba, Brazil, 2007, pp. 178-182.

[7] L. Likforman-Sulem, C. Faure, "*Extracting lines on handwritten documents by perceptual grouping"*, In Advances in Handwriting and drawing: a multidisciplinary approach, 1994, Europia, Paris, pp. 21-38.

[8] Y. Pu, Z. Shi, "*A natural learning algorithm based on Hough transform for text lines extraction in handwritten documents"*, In Proceedings of the 6th International Workshop on Frontiers in Handwriting Recognition, Taejon, Korea, 1998, pp. 637- 646.

[9] S. Nicolas, T. Paquet, and L. Heutte, *"Complex handwritten page segmentation using contextual model" In Proceedings of International Workshop on Document Image Analysis for Libraries*, Lyon, France, 2006, pp. 46-57.

[10] B. Gatos, A. Antonacopoulos, N. Stamatopoulos, *"Handwriting Segmentation Contest", 9th International Conference on Document Analysis and Recognition*, Curitiba, Brazil, 2007, pp. 1284-1288.