Pattern Recognition 42 (2009) 3158 – 3168

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A method for combining complementary techniques for document image segmentation

Nikolaos Stamatopoulos*, Basilis Gatos, Stavros J. Perantonis

Computational Intelligence Laboratory, Institute of Informatics and Telecommunications, National Center for Scientific Research "Demokritos", 153 10 Athens, Greece

ARTICLE INFO

Article history:
Received 8 August 2008
Accepted 15 October 2008

Keywords:
Document image segmentation
Combination method
Document image analysis
Segmentation

ABSTRACT

Image segmentation is a major task of handwritten document image 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 or font 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 line segmentation methods for handwritten documents demonstrate the effectiveness of the proposed combination method.

1. Introduction

Combining classifiers is a well researched topic in the pattern recognition community [1], for example in word recognition [2], spoken language processing [3] and biometric applications [4]. 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, linear and logistic regression.

In document analysis and recognition, several approaches have been proposed for improving OCR accuracy through combination [5]. These approaches can be categorized into two categories: (i) techniques in classifier combinations, as mentioned before, and (ii) string alignment combination methods [6,7]. Approaches of the second category combine several OCR outputs to produce a more accurate string estimate of the original text, but this cannot be done on character-by-character basis because of segmentations errors. Outputs strings must be aligned to extract an estimate and also errors must be uncorrelated. Furthermore, in Ref. [8], Ferrer et al. propose a method to improve the performance of individual page segmentation OCR engines based on the combination of the output of several engines. However, the rules of combination are designed after analyzing the results of each individual method and after visual inspection of the results.

Based on 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 [9–11]. A wide variety of methods have been proposed in the literature for document segmentation which can be categorized into five major categories: (1) projection profiles methods; (2) smearing methods; (3) methods based on the Hough transform; (4) grouping methods and (5) stochastic methods. Techniques from each category can confront some specific problems such as overlapping, touching components, image degradations, variability in skew angles and directions, disturbing elements, variability in inter-word and inter-character distances and others.

Projection profile methods [12–14] are among the most popular methods for document segmentation. They are commonly used for printed documents but can also be adapted to handwritten documents with little overlap. Moreover, these methods are not sensitive to writing fragmentation and they can handle inter-word and inter-character distances variability. However, projection profile methods cannot confront with the problem of skew and overlapping. For example, potential challenges encountered in text line segmentation task are the variable skew angles between different text lines on the same page and along the same text line and the presence of touching or overlapping words of two consecutive text lines.

Smearing methods, such as fuzzy RLSA (run-length smoothing algorithm) [15] and adaptive RLSA [16], are based on examining the white runs in a specified direction (usually vertical or horizontal)

* Corresponding author. Tel.: +30 210 6503218; fax: +30 210 653 2175.
E-mail addresses: nstam@iit.demokritos.gr (N. Stamatopoulos), bgat@iit.demokritos.gr, sper@iit.demokritos.gr
URL: http://www.iit.demokritos.gr/cil.

0031-3203/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2008.10.020

and usually followed by connected component analysis. Smearing methods enriched by local considerations can solve specific problems including overlapping and touching strokes. In addition, these methods work successfully with documents containing characters with variable font size. However, they can still have problems with variability in skew angles and directions. Also, smearing methods cannot handle the variability in inter-word and inter-character distances and finally, they usually use many thresholds and heuristic rules.

The Hough transform [17] is one of the most often used tools in many areas of document analysis. It has been applied for skew detection, string detection and line detection [18,19]. Hough transform can also be used to detect stroke orientation in handwritten words. It is a voting process where each pixel votes for all possible patterns (straight lines) passing through the point. The success of Hough transform can be explained by its global aspect, no a priori knowledge on point distribution is needed. In the line segmentation methods based on the Hough transform although the problem of variability in the skew angles between different text lines on the same page can be solved, problems with different skew angles along the same text line are difficult to handle.

Grouping methods [20,21] consist in building alignments by aggregating units in a bottom-up strategy. The units may be pixels or of higher level, such as connected components or blocks. Careful grouping rules make these methods convenient when they have to handle fluctuating data and variability in skew angles and directions. However, a wrong decision in an early stage of the grouping results in errors or incomplete alignments, for example, when lines, words or characters are very close.

Finally, stochastic methods [22,23] are more robust, but their implementation requires great care, particularly regarding the initialization phase. Also, these methods naturally avoid crossing overlapping components so they can generate non-linear segmentation paths to separate overlapping or touching components.

In this paper, we propose a combination method of complementary 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 and (ii) specific features of the initial document according to the segmentation problem which we have to solve, such as line, word or character segmentation. The rest of the paper organized as follows: In Section 2, the proposed combination method of different segmentation techniques is detailed. In Section 3, we present experimental results using two complementary line segmentation techniques for handwritten documents which demonstrate the effectiveness of the proposed method. Finally, conclusions are drawn in Section 4.

2. Combination method

This section describes the proposed combination method of different segmentation techniques. First, we give an overview of the method's steps. Afterwards, we give some definitions and finally we describe in detail the steps of the combination method.

2.1. General description

We consider that we have some different segmentation results of an initial image and combine them in order to increase the efficiency and the accuracy of the segmentation result. As subregions we define the regions which are created from the intersection of the segmentation results.

The combination method is composed of five steps:

Step 1: Average feature extraction.
Step 2: Detect correctly segmented regions.
Step 3: Divide subregions into groups.
Step 4: Create correctly segmented regions from each group.
Step 5: Final process of the new segmentation result.

In Step 1 we detect the subregions in which all the segmentation techniques are in agreement over a threshold (70%-high degree of overlap) and then we extract some features of each subregion. Finally, we calculate the average values of these features. Our goal is for the average features to approximate the average features of the correctly segmented regions of the initial image. For example, in text line segmentation task, it is expected that the features of the subregions approximate the features of text lines, such as length of line, height of line, etc (see Fig. 1).

In a similar way, in Step 2, we detect the subregions in which all the segmentation techniques are in agreement over a threshold (90%-very high degree of overlap). We decide that these subregions are correctly segmented regions and so we add them to the new segmentation result. Consequently, these subregions do not participate at the following steps. For example, in text line segmentation task, the detected subregions are text lines which have been detected correctly from all segmentation techniques (see Fig. 2).

In Step 3, we divide the remaining subregions into groups. We integrate into a group all the subregions which interrelate according

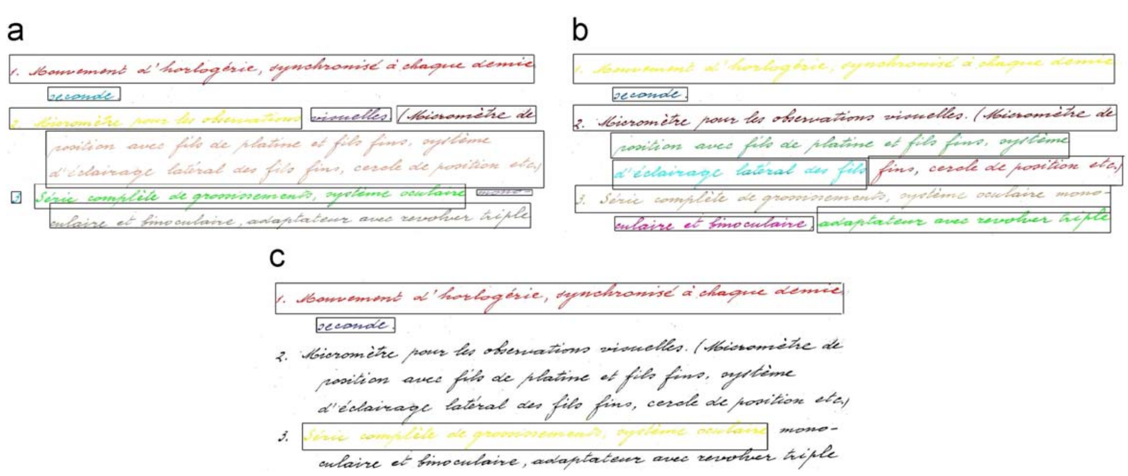

Fig. 1. An example of Step 1 in text line segmentation task; (a), (b) two different segmentation results with bounding boxes; (c) subregions from which we extract the features.

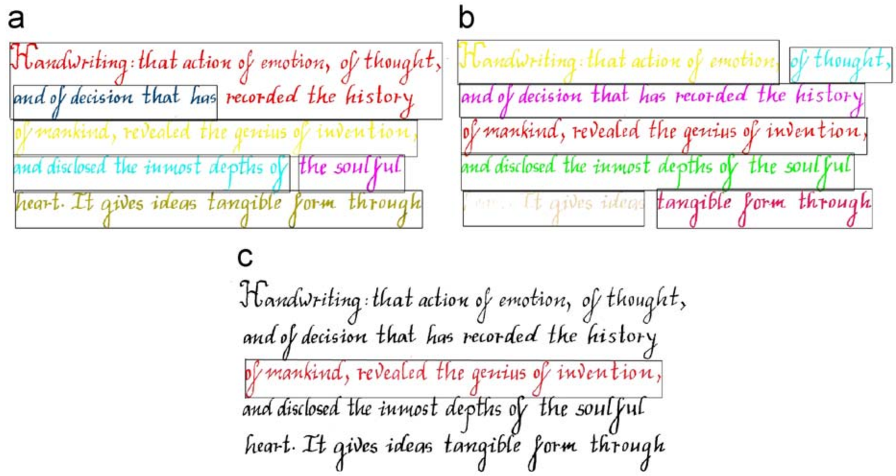

Fig. 2. An example of Step 2 in text line segmentation task; (a), (b) two different segmentation results with bounding boxes; (c) new segmentation result after applying Step 2.

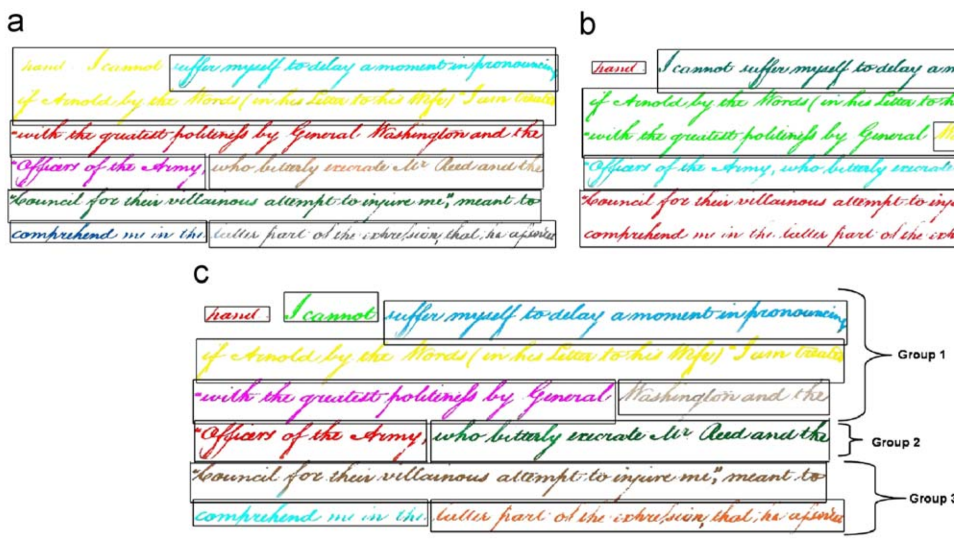

Fig. 3. An example of Step 3 in text line segmentation task; (a), (b) two different segmentation results with bounding boxes; (c) groups of subregions after applying Step 3.

to segmentation results. We want all the subregions which form a correctly segmented region to be members of the same group. So, a group contains the subregions which form one or more correctly segmented regions. For example, in text line segmentation task, a group contains the subregions which form one or more text lines. In other words, the subregions of a text line should not belong to different groups (see Fig. 3). Our goal is to examine, at the next step, each group separately and decide which subregions of it will be merged in order to create correctly segmented regions.

In Step 4, we process every group of the previous step independently (see Fig. 4). We start from a subregion of the group which has the highest degree of overlap according to the segmentation results. Then, we examine which others subregions should be merged with it until the features of the new region are closer to the average features of Step 1. So the new region is accounted as a correctly segmented region and we add it to the new segmentation result. We repeat this process until all the subregions of the group have been added to the new segmentation result.

Finally, in Step 5, we detect all the pixels of the foreground of the initial which have not been added to the new segmentation result and we merge them with the nearest region.

2.2. Definitions

Definition 1. Initial binary image

We consider a binary image:

$$I(x,y)=\begin{cases}1 \\ 0\end{cases} \quad \text{where } 1\leqslant x\leqslant I_x,\ 1\leqslant y\leqslant I_y \tag{1}$$

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

Definition 2. Segmentation results

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

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

Each value r_j in the set A_j denotes that a pixel belongs to the r_j^{th} segment according to the jth method of segmentation. In Fig. 5 we see a specific example with $N=3$ segmentation methods.

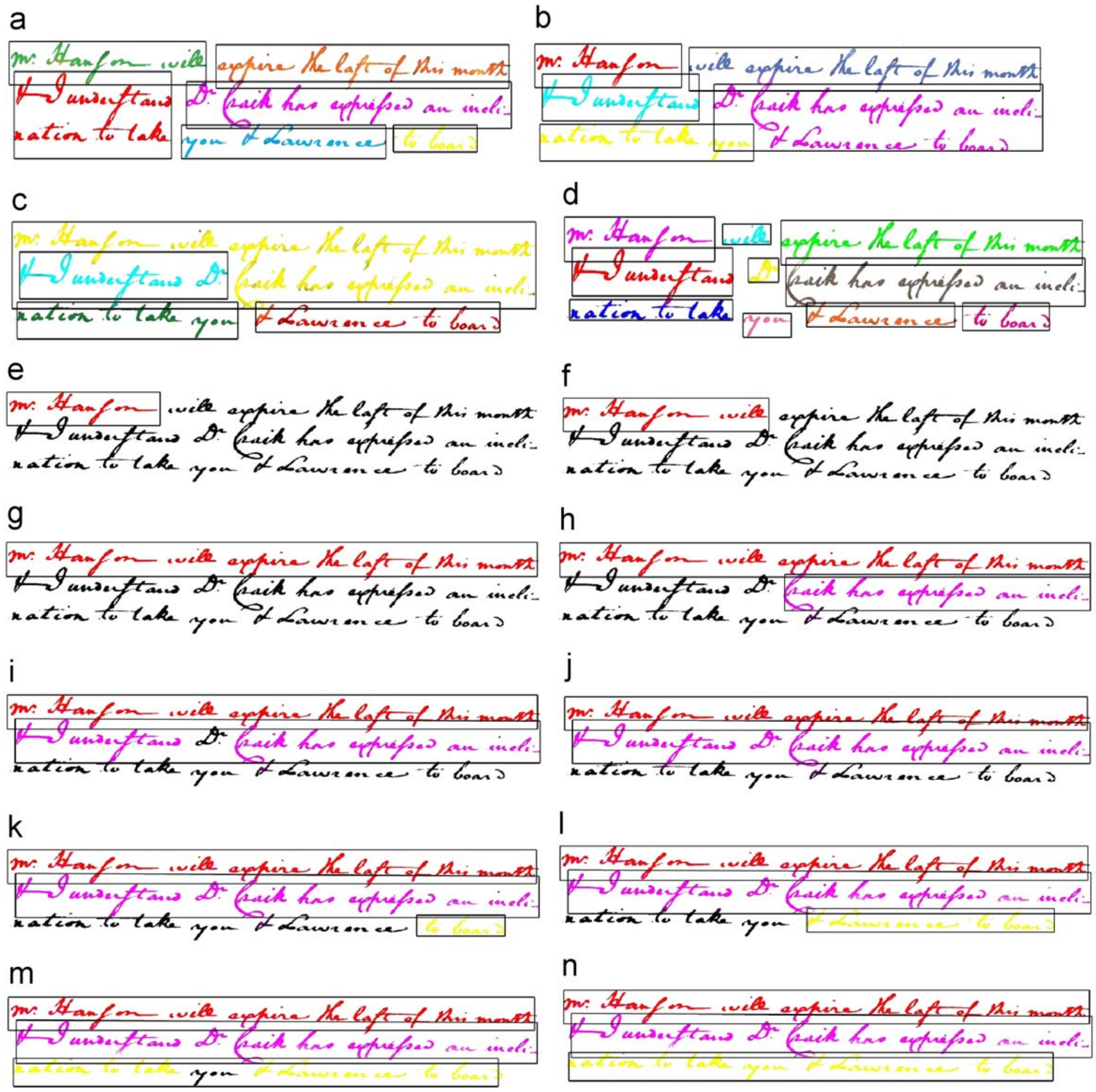

Fig. 4. An example of Step 4 in text line segmentation task; (a)–(c) three different segmentation results with bounding boxes; (d) subregions which belong at the same group; (e)–(g) process of creation of the first correctly segmented region; (h)–(j) process of creation of the second correctly segmented region; (k)–(n) process of creation of the third correctly segmented region.

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

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

using the segmentation results $R_1(x,y), R_2(x,y), \ldots, R_N(x,y)$.

Definition 3. Intersection of the segmentation results

We define the following binary images:

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

where $1 \leqslant r_j \leqslant n_j, j=1,\ldots,N$, which represents the intersection of the segmentation results with segment ids $(r_1,r_2,\ldots,r_N)$ and so it defines the subregion with segment ids $(r_1,r_2,\ldots,r_N)$ (see Fig. 5e), and

$$D_j(x,y)_{(r_j)} = \begin{cases} 1 & \text{if } R_j(x,y)=r_j \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $1 \leqslant r_j \leqslant n_j, j=1,\ldots,N$, which represents the intersection of the image $I(x,y)$ with the r_jth segment according to the jth method of segmentation (see Fig. 5f).

Definition 4. Overlap between the segmentation results

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

$$f_j(r_1,r_2,\ldots,r_N) = \begin{cases} \frac{\sum_{x,y} C(x,y)_{(r_1,r_2,\ldots,r_N)}}{\sum_{x,y} D_j(x,y)_{(r_j)}} & \text{if } \sum_{x,y} D_j(x,y)_{(r_j)} \neq 0 \\ 0 & \text{othewise} \end{cases} \tag{6}$$

where $1 \leqslant r_j \leqslant n_j, j=1,\ldots,N$.

For example, in Fig. 5,

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

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 $R_3(x,y)$.

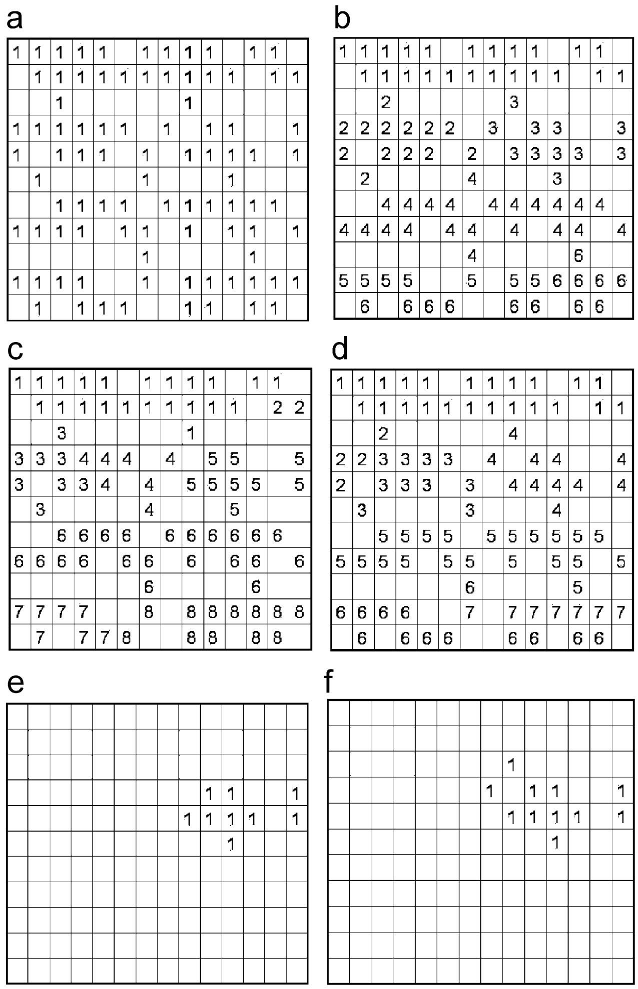

Fig. 5. An example with three different segmentation results, where blank pixels represent the pixels of background (a) binary image $I(x,y)$; (b), (c), (d) segmentation results $R_1(x,y)$, $R_2(x,y)$ and $R_3(x,y)$, respectively; (e) binary image $C(x,y)_{(3,5,4)}$; (f) binary image $D_1(x,y)_{(3)}$.

Definition 5. High and very high degree of overlap

Using the function $f_j(r_1,r_2,\ldots,r_N)$ we define the sets HO and VHO of subregions with segment ids $(r_1,r_2,\ldots,r_N)$. The set HO contains the subregions with segment ids $(r_1,r_2,\ldots,r_N)$ in which all the segmentation results have high degree of overlap ($f_j(r_1,r_2,\ldots,r_N)\geqslant 70\%$) and is defined as follows:

$$(r_1,r_2,\ldots,r_N) \in HO \quad \text{if } f_j(r_1,r_2,\ldots,r_N)\geqslant 0.7 \quad \forall j=1,\ldots,N \tag{7}$$

Similarly, the set VHO contains the subregions with segment ids $(r_1,r_2,\ldots,r_N)$ in which all the segmentation results have very high degree of overlap ($f_j(r_1,r_2,\ldots,r_N)\geqslant 90\%$) and is defined as follows:

$$(r_1,r_2,\ldots,r_N) \in VHO \quad \text{if } f_j(r_1,r_2,\ldots,r_N)\geqslant 0.9 \quad \forall j=1,\ldots,N \tag{8}$$

Definition 6. Feature extraction

Finally, we define the following function:

$$V[Q(x,y)] = \mathit{Feature\,Extraction}\,[Q(x,y)] \tag{9}$$

which receives as input a binary image $Q(x,y)$ and returns a vector $V[Q(x,y)]=\{v_1,v_2,\ldots,v_p\}$, where $0\leqslant v_i\leqslant 1, i=1,\ldots,p$ representing specific features of the image $Q(x,y)$. We choose the features according to the segmentation problem which we have to solve, such as line segmentation, word segmentation etc. In Section 3 we describe specific features in order to combine line segmentation methods.

2.3. Methodology

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

Step 1: Average feature extraction.

At this step we extract the features of each subregion which is member of set HO (see Eq. (7)) and then we calculate the average values of these features $AV=\{av_1,av_2,\ldots,av_p\}$. In this way, it is expected that the vector AV approximates the average features of the correctly segmented regions. In order to achieve that we use the following algorithm:

Algorithm 1. *Calculate_Vector_AV* [ ]

{
Step_1: *Extract the features of each subregion which is member of set HO*
$\forall (r_1,r_2,\ldots,r_N) \in HO$
$V[C(x,y)_{(r_1,r_2,\ldots,r_N)}] = \mathit{FeatureExtraction}[C(x,y)_{(r_1,r_2,\ldots,r_N)}]$
Step_2: *Calculate the average features*
$AV = \frac{\sum_{i=1}^{m} V[C(x,y)_{(r_1,r_2,\ldots,r_N)}]}{m}$
}

where

$$m=|HO| \tag{10}$$

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

Step 2: Detect correctly segmented regions.

At this step we add to the new segmentation result $R(x,y)$ the subregions which are members of set VHO (see Eq. (8)). These subregions are accounted as correctly segmented regions and do not participate at the following steps. In order to achieve that we use the following algorithm:

Algorithm 2. *Detect_Correctly_Segmented_Regions* [ ]

{
Step_1: *Initialization of new segmentation result*
$R(x,y)=0 \quad \forall x,y$
Step_2: *Add subregions which are members of set VHO to the new segmentation result*
$k=0$
$\forall (r_1,r_2,\ldots,r_N) \in VHO$
{
$k=k+1$
$R(x,y)=k^{*}C(x,y)_{(r_1,r_2,\ldots,r_N)} \quad \forall x,y$
}
}

where $k=1,\ldots,z$ and

$$z=|VHO| \tag{11}$$

Looking at the example of Fig. 5, we can detect the subregions with segment ids (1, 1, 1) and (4, 6, 5) (see Figs. 6 and 7) which are

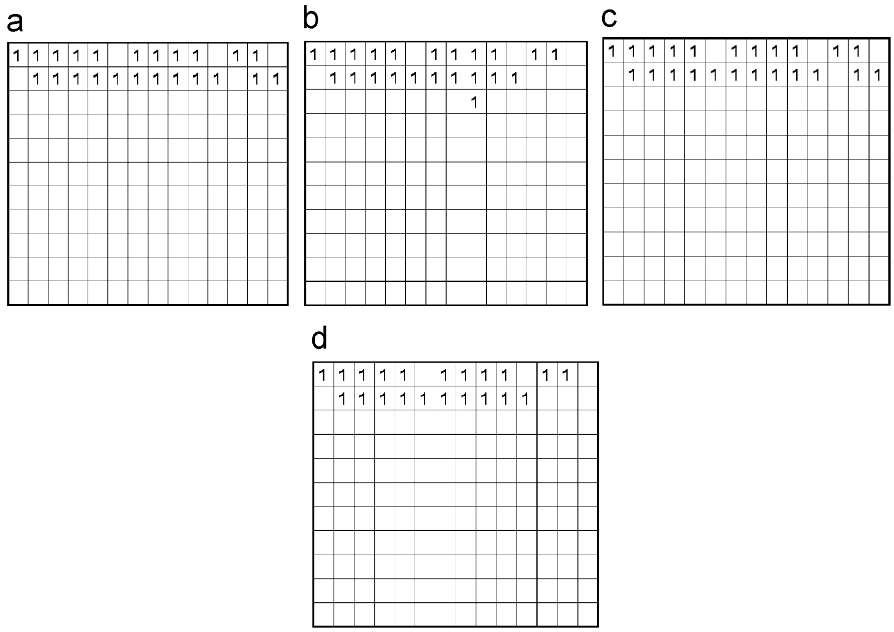

Fig. 6. Subregion with segment ids (1,1,1) which is member of set *VHO* at the example of Fig. 5 (a)–(c); binary images $D_1(x,y)_{(1)}$, $D_2(x,y)_{(1)}$, $D_3(x,y)_{(1)}$ respectively; (d) binary image $C(x,y)_{(1,1,1)}$.

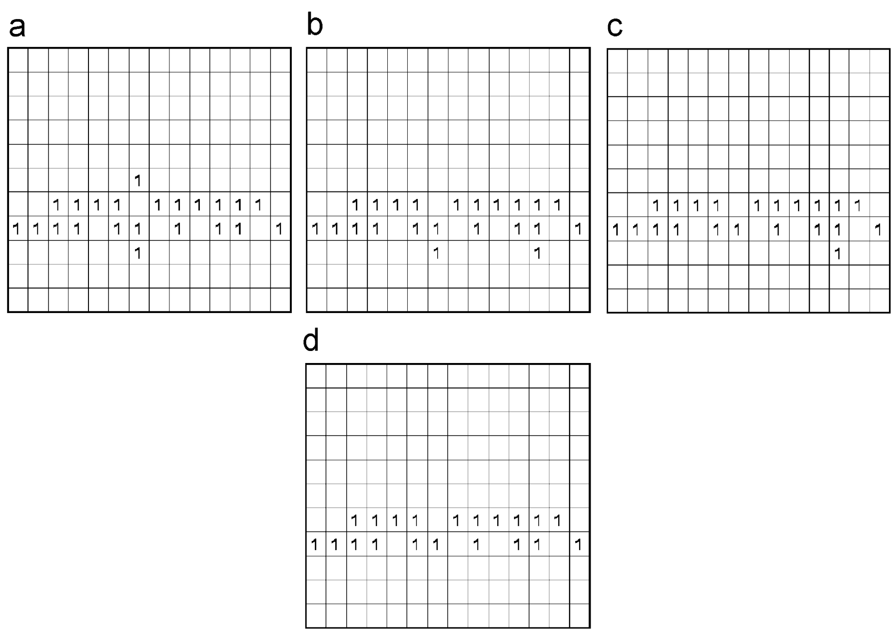

Fig. 7. Subregion with segment ids (4,6,5) which is member of set *VHO* at the example of Fig. 5 (a)–(c); binary images $D_1(x,y)_{(4)}$, $D_2(x,y)_{(6)}$, $D_3(x,y)_{(5)}$, respectively; (d) binary image$C(x,y)_{(4,6,5)}$.

members of set *VHO*:

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

$$f_3(1,1,1) = \frac{21}{23} = 0.91$$

and

$$f_1(4,6,5) = \frac{20}{22} = 0.90, \quad f_2(4,6,5) = \frac{20}{22} = 0.90,$$

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

Fig. 8 depicts the new segmentation result $R(x,y)$ after applying Step 2 to the example of Fig. 5.

Step 3: Divide subregions into groups.

At this step the subregions with segments ids $(r_1, r_2, \ldots, r_N)$, which have been used in the previous step or they have at least one common segment id $r_1, r_2, \ldots, r_N$ with them, do not participate. In our example, the subregions which do not participate are the following: $(1,1,1)$ and $(4,6,5)$ because they have been used in Step 2 (see Figs. 6 and 7) and also the subregions $(1,2,1), (3,1,4), (4,4,3), (4,6,6)$ and $(6,6,5)$ since they have at least one common segment id with the combinations which have been used (see Fig. 9).

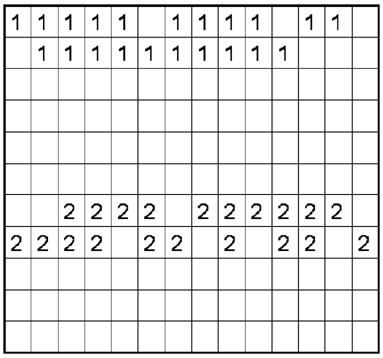

Fig. 8. New segmentation result $R(x,y)$ after applying Step 2 to the example of Fig. 5.

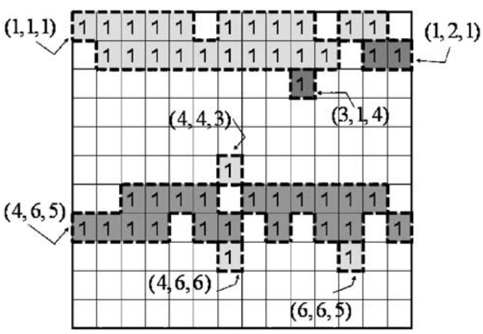

Fig. 9. Subregions which do not participate at Step 3.

We divide the remaining subregions with segments ids $(r_1, r_2, \ldots, r_N)$, where $\sum_{x,y} C(x,y)_{(r_1,r_2,\ldots,r_N)} \neq 0$, into groups G_i, $i=1,\ldots,l$. We want all the subregions which form a correctly segmented region to be members of the same group. In other words, a group contains the subregions which form one or more correctly segmented regions. The general objective is to examine, at the next step, each group separately and decide which subregions of it will be merged in order to create correctly segmented regions. In order to create the groups we use the following algorithm:

Algorithm 3. *Create_Groups* [ ]

{
  $i=0$
  Until all subregions $(r_1, r_2, \ldots, r_N)$: $\sum_{x,y} C(x,y)_{(r_1,r_2,\ldots,r_N)} \neq 0$
  have joined a group
  {
    Step_1: *Create a new empty group* G_i
      $i=i+1$
      $G_i=\emptyset$
    Step_2: *Find a subregion which doesn't belong to a group*
      Find $(r_1, r_2, \ldots, r_N)$: $[\sum_{x,y} C(x,y)_{(r_1,r_2,\ldots,r_N)} \neq 0$ *AND*
      $(r_1, r_2, \ldots, r_N) \notin G_j, j=1,\ldots,i-1]$
    Step_3: *Call the recursive function Join_Group*
      Join_Group$[G_i,(r_1,r_2,\ldots,r_N)]$
  }
}

where the recursive function *Join_Group*$[G_i,(r_1,r_2,\ldots,r_N)]$, which receives as input a group and a subregion which doesn't belong to a group, is defined as follows:

Algorithm 4. *Join_Group* $[G_i, (r_1, r_2, \ldots, r_N)]$

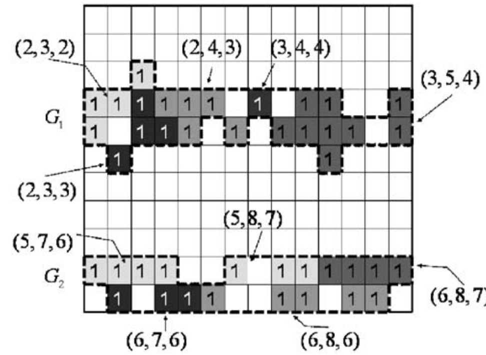

Fig. 10. Groups G_1 and G_2 after applying Step 3.

{
  Step_1: *Add the new subregion to the group* G_i
    $G_i = G_i \cup \{(r_1, r_2, \ldots, r_N)\}$
  Step_2: *Find all the subregions which have at least one common segment id with the initial subregion and they don't belong to a group*
    $\forall (r'_1, r'_2, \ldots, r'_N)$: $[\sum_{x,y} C(x,y)_{(r'_1,r'_2,\ldots,r'_N)} \neq 0$ *AND*
    $(r'_1, r'_2, \ldots, r'_N) \notin G_j, j=1,\ldots,i-1]$
      if $(r_1 = r'_1$ *OR* … *OR* $r_N = r'_N)$
        Step_3: *Call the recursive function Join_Group*
          Join_Group$[G_i = (r'_1, r'_2, \ldots, r'_N)]$
}

In our example (see Fig. 5), two groups will be created:

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

and

$G_2 = \{(5,7,6),(5,8,7),(6,7,6),(6,8,6),(6,8,7)\}$

Fig. 10 depicts these groups and their subregions.

Step 4: Create correctly segmented regions from each group.

At this step we process every group $G_i, i=1,\ldots,l$ independently. Algorithm 5 (*Process_Group*) finds a subregion with segments ids $(r_1, r_2, \ldots, r_N)$ in which the segmentation results have the highest degree of overlap according to one segmentation result and also it has not been added to the new segmentation result ($g((r_1, r_2, \ldots, r_N))=0$, see Eq. (12)). Then, using Algorithm 6 *(Merge_Subregions)*, it examines which others subregions should be merged with it until the features of the new region are closer to the average features *AV* of Step 1. So the new region is accounted as a correctly segmented region and we add it to the new segmentation result. We repeat this process until all the subregions of the group have been added to the new segmentation result. Algorithm 5 (*Process_Group*) is defined as follows:

Algorithm 5. *Process_Group* $[G_i]$

{
  Step_1: *Initialize function g and parameters k and STOP*
    $\forall (r_1, r_2, \ldots, r_N) \in G_i \quad g((r_1, r_2, \ldots, r_N)) = 0$
    k = z *//see Eq.* (11)
    STOP = false
  Step_2: *Stop when every subregion of the group has been added to the new segmentation result*
    Until (*STOP = true*)
    {
      $k=k+1$

Step_3: *Find the subregion which has the highest degree of overlap*
$MI = \max_{\substack{(r_1,r_2,\ldots,r_N)\in G_i \\ g((r_1,r_2,\ldots,r_N))=0}} \{f_1(r_1,r_2,\ldots,r_N),\ldots,f_N(r_1,r_2,\ldots,r_N)\}$
Find $(r_1,r_2,\ldots,r_N)\in G_i:\ [g((r_1,r_2,\ldots,r_N))=0$ *AND* $(f_1(r_1,r_2,...,r_N)=MI$ *OR....OR* $f_N(r_1,r_2,...,r_N)=MI)]$
If $(MI=0)$
STOP = true
Else
{
Step_4: *Add this subregion to the new segmentation result*
$R(x,y)=k^*C(x,y)_{(r_1,r_2,\ldots,r_N)} \quad \forall\, x,y$
$g((r_1,r_2,\ldots,r_N))=1$
Step_5: *Calculate the difference between the features of it and the average features of the image*
$V[C(x,y)_{(r_1,r_2,\ldots,r_N)}] = FeatureExtraction[C(x,y)_{(r_1,r_2,\ldots,r_N)}]$
$Dif=|AV-V[C(x,y)_{(r_1,r_2,\ldots,r_N)}]|$
$M(x,y)=C(x,y)_{(r_1,r_2,\ldots,r_N)} \quad \forall\, x,y$
Step_6: *Examine if we should add other subregions*
Merge[G_i, *Dif*, *M*(*x*,*y*), *k*]
}
}
}

where

$$g((r_1,r_2,\ldots,r_N)) = \begin{cases} 1 & \text{if } (r_1,r_2,\ldots,r_N) \text{ has been added to the} \\ & \quad \text{new segmentation result} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

and function *Merge_Subregions*$[G_i, Dif, M(x,y), k]$, which examine which others subregions should be merged with the initial subregion, is defined as follows:

Algorithm 6. *Merge_Subregions* [G_i, *Dif*, *M*(*x*,*y*), *k*]

{
Step_1: *Initialize parameter STOP*
STOP = false
Step_2: *Stop when we have examined every subregion of the group*
Until (*STOP = true*)
{
Step_ 3: *Find the subregion which has the highest degree of overlap and it has not been examined*
$MI = \max_{\substack{(r_1,r_2,\ldots,r_N)\in G_i \\ g((r_1,r_2,\ldots,r_N))=0}} \{f_1(r_1,r_2,\ldots,r_N),\ldots,f_N(r_1,r_2,\ldots,r_N)\}$
Find $(r_1,r_2,\ldots,r_N)\in G_i:\ [g((r_1,r_2,\ldots,r_N))=0$ *AND* $(f_1(r_1,r_2,...,r_N)=MI$ *OR....OR* $f_N(r_1,r_2,...,r_N)=MI)]$
If $(MI=0)$
STOP = true
Else
{
Step_4: *Merge this subregion with the initial region M*(*x*,*y*)
$N(x,y)=M(x,y)+C(x,y)_{(r_1,r_2,\ldots,r_N)} \quad \forall\, x,y$
$g((r_1,r_2,\ldots,r_N))=1$
Step_5: *Calculate the difference between the features of it and the average features of the image*
$V[N(x,y)] = Feature\,Extraction[N(x,y)]$
$Dif_1=|AV-V[N(x,y)]|$
If $(Dif_1 < Dif)$
{
Step_6: *Add the new subregion to the new segmentation result*
$R(x,y)=k^*C(x,y)_{(r_1,r_2,\ldots,r_N)} \quad \forall\, x,y$
$M(x,y)=N(x,y) \quad \forall\, x,y$
}
Else
$g((r_1,r_2,\ldots,r_N))=0$
}
}
}

Fig. 11 depicts the distinct steps which we follow in order to process group $G_2=\{(5,7,6),(5,8,7),(6,7,6),(6,8,6),(6,8,7)\}$ in our example, (see Fig. 10):

(1) Algorithm 5 (Process_Group):

- Start from subregion with segment ids $(5,7,6)$ which has $f_1(5,7,6)=4/7=0.57$ (see Fig. 11a).
- Add it to the new segmentation result *R*(*x*,*y*) (see Fig. 11b).
- Calculate the difference between the features of it and average features *AV*, let $Dif=0.9$.

(2) Algorithm 6 (Merge_Subregions):

- Examine if we can merge with it the subregion with segment ids $(6,8,7)$ which has $f_3(6,8,7)=4/7=0.57$, so we calculate the difference between the features of the new region and average features *AV*, let $Dif_2=0.7$, (see Fig. 11c).
- $Dif_2 < Dif_1$, so we add it to the new segmentation result *R*(*x*,*y*) (see Fig. 11d).
- Examine if we can merge with it the subregion with segment ids $(5,8,7)$ which has $f_1(5,8,7)=3/7=0.43$, so we calculate the difference between the features of the new region and average features *AV*, let $Dif_3=0.5$, (see Fig. 11e).
- $Dif_3 < Dif_2$, so we add it to the new segmentation result *R*(*x*,*y*) (see Fig. 11f).
- Examine if we can merge with it the subregion with segment ids $(6,7,6)$ which has $f_2(6,7,6)=3/7=0.43$ (see Fig. 11(g), so we calculate the difference between the features of the new region and average features *AV*, let $Dif_4=0.6$.
- $Dif_4 > Dif_3$, so we do not add it to the new segmentation result *R*(*x*,*y*).
- Examine if we can merge with it the subregion with segment ids $(6,8,6)$ which has $f_1(6,8,6)=5/12=0.41$ (see Fig. 11(h), so we calculate the difference between the features of the new region and average features *AV*, let $Dif_5=0.55$.
- $Dif_5 > Dif_3$, so we do not add it to the new segmentation result *R*(*x*,*y*).

(3) Algorithm 5 (Process_Group):

- We have examined all the subregions, so we start now from subregion with segment ids $(6,7,6)$ which has $f_2(6,7,6)=3/7=0.43$ (see Fig. 11i).
- Add it to the new segmentation result *R*(*x*,*y*) (see Fig. 11j).
- Calculate the difference between the features of it and average features *AV*, let $Dif_6=0.85$.

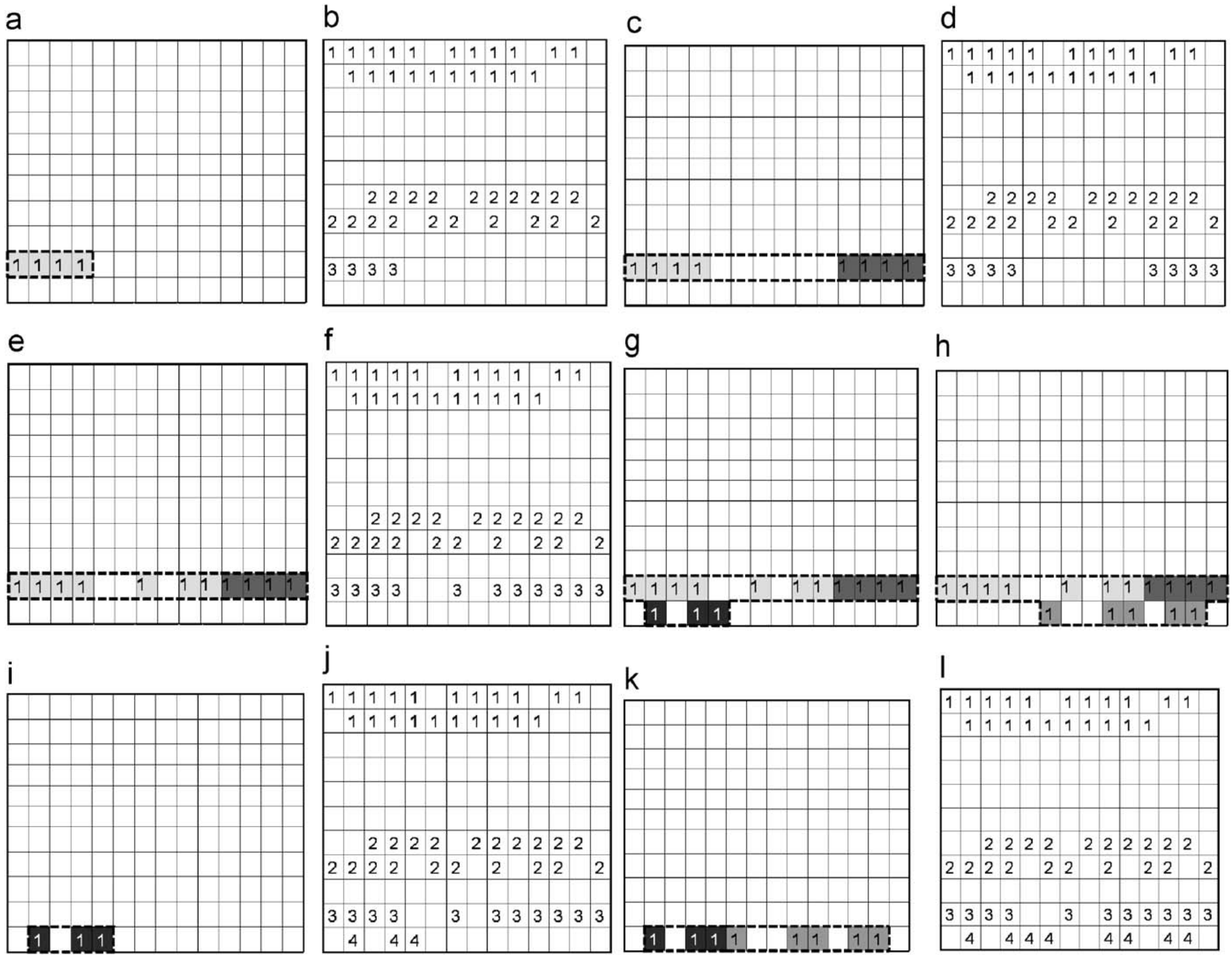

Fig. 11. Apply Step 4 at group G_2.

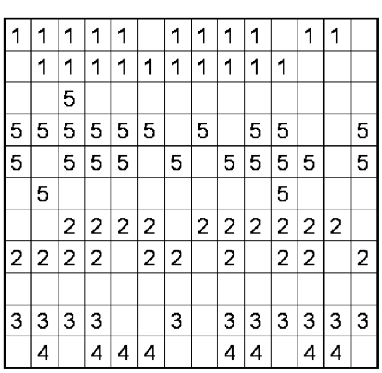

Fig. 12. New segmentation result $R(x,y)$ after applying Step 4 at group G_1.

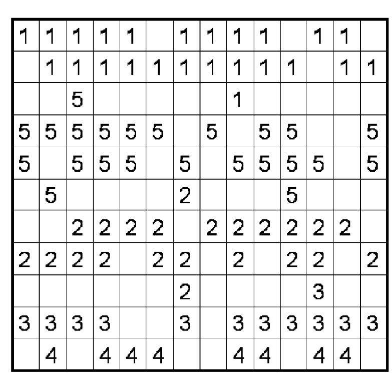

Fig. 13. Final segmentation result $R(x,y)$ after applying Step 5.

(4) Algorithm 6 (Merge_Subregions):

- Examine if we can merge with it the subregion with segment ids $(6,8,6)$ which has $f_1(6,8,6)=5/12=0.41$, so we calculate the difference between the features of the new region and average features AV, let $Dif_7=0.45$, (see Fig. 11k).
- $Dif_7 < Dif_6$, so we add it to the new segmentation result $R(x,y)$ (see Fig. 11(l)).

With similar steps, we process group G_1 (see Fig. 12).

Step 5: Final process of the new segmentation result.

At this step every pixel of the foreground, which does not have a segment id at the new segmentation result image $R(x,y)$

Table 1
Comparative results

	M	o2o	gt_o2m	gt_m2o	d_o2m	d_m2o	DR (%)	RA (%)	FM (%)
Projection profiles	1248	841	18	163	72	38	77.4	69.5	73.2
Adaptive RLSA	1314	860	71	168	78	156	80.3	69.9	74.7
After combination of projection profiles and adaptive RLSA	1152	1071	11	52	26	25	94.9	94.0	94.4

(see Fig. 9), inherits the id of the majority of its eight neighbors pixels. Fig. 13 depicts the final segmentation result $R(x,y)$ after applying Step 5 in our example.

3. Experimental results

To verify the validity of the proposed method we use two complementary line segmentation methods, projection profiles based on Ref. [12] and Adaptive RLSA based on Ref. [16]. In Ref. [12], each minimum of the profile curve is a potential segmentation point. Potential points are then scored according to their distance to adjacent segmentation points. The reference distance is obtained from the histogram of distances between adjacent potential segmentation points. The highest scored segmentation point is used as an anchor to derive the remaining ones. In Ref. [16], Makridis and Nikolaou propose the adaptive RLSA which is an extension of the classical RLSA in the sense that additional smoothing constraints are set in regard to the geometrical properties of neighboring connected components. The replacement of background pixels with foreground pixels is performed when these constraints are satisfied.

We apply each method to a set of 50 handwritten documents (1144 text line segments), 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 combination method. For this reason we use the following features of a region according to Eq. (9):

$$v_1 = \frac{\text{length of the bounding box}}{\text{length of image}} \quad (13)$$

$$v_2 = \frac{\text{height of the bounding box}}{\text{height of image}} \quad (14)$$

$$v_3 = \frac{v_2}{v_1} \quad (15)$$

$$v_4 = \frac{\text{foreground pixels}}{\text{total pixels}} \quad (16)$$

$$v_5 = \frac{x \text{ co-ordinate of middle point}}{\text{length of image}} \quad (17)$$

$$v_6 = \frac{\text{median stroke width in the row with the maximum number of black–white transitions}}{\text{length of image}} \quad (18)$$

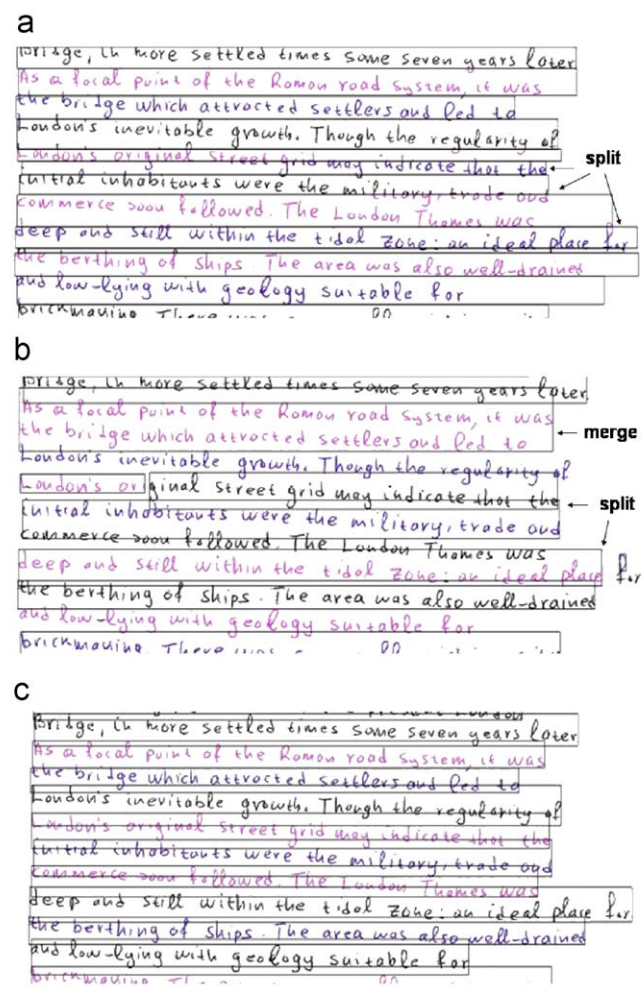

Fig. 14. Segmentation results with the bounding boxes: (a) projection profiles; (b) adaptive RLSA; (c) final result after combination.

For the purpose of the evaluation, we manually marked the correct line segments in the set of 50 images. The performance evaluation was based on counting the number of matches between the lines detected by the segmentation algorithms or their combination and the lines in the ground truth [16,24,25]. We used a MatchScore table whose values are calculated according to the overlap of the labeled pixel set as text line and the ground truth. Let I be the set of all images points, G_j the set of all points inside the j ground truth region, R_i the set of all points inside the i result region, $T(s)$ a function that counts the elements of set s. Table $MatchScore(i,j)$ represents the matching results of the j ground truth region and the i result region. Based on a pixel based approach of Ref. [25], we define that:

$$Match\,Score(i,j) = \frac{T(G_j \cap R_i \cap I)}{T((G_j \cup R_i) \cap I)} \quad (19)$$

We consider a one-to-one match only if the matching score is equal or above the evaluator's acceptance threshold T_a. If N is the count of ground truth regions, M is the count of result regions and w_1, w_2,w_3,w_4,w_5,w_6 are pre-determined weights, we calculate the detection rate (DR) and recognition accuracy (RA) as follows:

$$DR = w_1 \frac{o2o}{N} + w_2 \frac{gt\_o2m}{N} + w_3 \frac{gt\_m2o}{N} \quad (20)$$

$$RA = w_4 \frac{o2o}{M} + w_5 \frac{d\_o2m}{M} + w_6 \frac{d\_m2o}{M} \quad (21)$$

where $o2o$ (one-to-one), gt_o2m (one-to-many), gt_m2o (many-to-one), d_o2m and o_m2o are calculated from MatchScore table (see Eq. (19)) using acceptance threshold T_a and following the steps of Ref. [24].

An overall F-measure (FM) for text line detection can be defined if we combine the values of detection rate (DR) and recognition accuracy (RA):

$$FM = \frac{2DR \quad RA}{DR + RA} \tag{22}$$

We evaluated the performance of the two segmentation algorithms and their combination for text line using Eqs. (19)–(22) for all 50 images with $N = 1144$ text line segments and parameters $w_1 = 1$, $w_2 = 0.25$, $w_3 = 0.25$, $w_4 = 1$, $w_5 = 0.25$, $w_6 = 0.25$ and acceptance threshold $T_a = 95\%$. As depicted in Table 1, the new segmentation result outperforms the two others methods and it increases the overall evaluation measure about 20%. Fig. 14 depicts an example of the proposed combination method, in which after the combination method all text lines have been detected correctly. Even though both segmentation methods, projection profiles and Adaptive RLSA, have splitted the same text line, this has been corrected by the combination method.

4. Concluding remarks

This paper proposes an efficient combination method of segmentation techniques. The proposed method combines the segmentation results of complementary techniques, where each technique can solve some different difficult problems, 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. Experimental results on line segmentation methods demonstrate the effectiveness of the proposed combination method, as it increases the overall evaluation measure about 20%. 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.

Acknowledgement

The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement no 215064 (project IMPACT).

References

[1] J. Kittler, M. Hatef, R. Duin, J. Matas, On combining classifiers, IEEE Trans. Pattern Anal. Mach. Intell. 20 (3) (1998) 226–239.
[2] A.L. Koerich, R. Sabourin, C.Y. Suen, Recognition and verification of unconstrained handwritten words, IEEE Trans. Pattern Anal. Mach. Intell. 27 (10) (2005) 1509–1522.
[3] M. Magimai-Doss, D. Hakkani-Tur, O. Cetin, E. Shriberg, J. Fung, N. Mirghafori, Entropy based classifier combination for sentence segmentation, in: IEEE International Conference on Acoustics, Speech and Signal Processing, Honolulu, Hawaii, 2007, pp. 189–192.
[4] S. Tulyakov, V. Govindaraju, Classifier combination types for biometric applications, 2006, in: Conference on Computer Vision and Pattern Recognition Workshop, New York, pp. 58.
[5] J.C. Handley, Improving OCR accuracy through combination: a survey, in: International Conference on Systems, Man, and Cybernetics, California, USA, 1998, pp. 4330–4333.
[6] J.C. Handley, T.B. Hickey, Merging three optical character recognition outputs for improved precision using a minimum edit distance function, OCLC Online Computer Library, US Patent No. 5459739, 1995.
[7] S.V. Rice, J. Kanai, T.A. Nartker, An algorithm for matching OCR-generated text strings, Int. J. Pattern Recognition Artif. Intell. 8 (5) (1994) 1259–1268.
[8] M. Ferrer, E. Valveny, Combination of OCR engines for page segmentation based on performance evaluation, in: International Conference on Document Analysis and Recognition, Curitiba, Brazil, September 2007, pp. 784–788.
[9] L. Likforman-Sulem, A. Zahour, B. Taconet, Text line segmentation of historical documents: a survey, Int. J. Doc. Anal. Recognition 2–4 (2007) 123–138.
[10] G. Seni, E. Cohen, External word segmentation of off-line handwritten text lines, Pattern Recognition 27 (1) (1994) 41–52.
[11] R.G. Casey, E. Lecolinet, A survey of methods and strategies in character segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 18 (7) (1996) 690–706.
[12] A. Antonacopoulos, D. Karatzas, Document image analysis for World War II personal records, in: First International Workshop on Document Image Analysis for Libraries, Palo Alto, 2004, pp. 336–341.
[13] M. Arivazhagan, H. Srinivasan, S.N. Srihari, A statistical Approach to handwritten line segmentation, in: Document Recognition and Retrieval XIV, Proceedings of SPIE, San Jose, CA, February 2007, pp. 6500T-1–11.
[14] R. Manmatha, J.L. Rothfeder, A scale space approach for automatically segmenting words from historical handwritten documents, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005) 1212–1225.
[15] Z. Shi, V. Govindaraju, Line separation for complex document images using fuzzy runlength, in: First International Workshop on Document Image Analysis for Libraries, 2004, pp. 306.
[16] B. Gatos, A. Antonacopoulos, N. Stamatopoulos, ICDAR2007 handwriting segmentation contest, in: Ninth International Conference on Document Analysis and Recognition, Curitiba, Brazil, September 2007, pp. 1284–1288.
[17] P.V.C. Hough, Methods and means for recognizing complex patterns, US patent 3069654, 1962.
[18] Y. Pu, Z. Shi, A natural learning algorithm based on Hough transform for text lines extraction in handwritten documents, in: Proceedings of the Sixth International Workshop on Frontiers in Handwriting Recognition, Taejon, Korea, 1998, pp. 637–646.
[19] G. Louloudis, B. Gatos, C. Halatsis, Text line detection in unconstrained handwritten documents using a block-based Hough transform approach, in: Conference on Document Analysis and Recognition, Curitiba, Brazil, September 2007, pp. 599–603.
[20] L. Likforman-Sulem, C. Faure, Extracting Lines on Handwritten Documents by Perceptual Grouping, in: Advances in Handwriting and Drawing: a Multidisciplinary Approach, Europia, Paris, 1994, pp. 21–38.
[21] M. Feldbach, K.D. Tonnies, Line detection and segmentation in historical church registers, in: Sixth International Conference on Document Analysis and Recognition, Seattle, 2001, pp. 743–747.
[22] S. Nicolas, T. Paquet, 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.
[23] Y.H. Tseng, H.J. Lee, Recognition-based handwritten Chinese character segmentation using a probabilistic Viterbi algorithm, Pattern Recognition Lett. 20 (8) (1999) 791–806.
[24] I. Phillips, A. Chhabra, Empirical performance evaluation of graphics recognition systems, IEEE Trans. Pattern Anal. Mach. Intell. 21 (9) (1999) 849–870.
[25] B.A. Yanikoglu, L. Vincent, Pink Panther: a complete environment for ground-truthing and benchmarking document page segmentation, Pattern Recognition 31 (9) (1994) 1191–1204.

About the Author—NIKOLAOS STAMATOPOULOS was born in Athens in 1984. He graduated in 2006 from the Department of Informatics and Telecommunications of National and Kapodistrian University of Athens, and from November 2006 he is a Ph.D. candidate at the same university. Currently, he is working on the "IMPACT" project, at the Computational Intelligence Laboratory, Institute of Informatics and Telecommunications of the National Center for Scientific Research "Demokritos". His research interests are in Image Processing and Document Image Analysis, Processing of Historical Documents.

About the Author—BASILIS GATOS received his Electrical Engineering Diploma in 1992 and his Ph.D. degree in 1998, both from the Electrical and Computer Engineering Department of Democritus University of Thrace, Xanthi, Greece. He is currently working as a Researcher at the Institute of Informatics and Telecommunications of the National Center for Scientific Research "Demokritos", Athens, Greece. His main research interests are in Image Processing, Pattern Recognition, Document Image Analysis, OCR, Processing and Recognition of Historical Documents.

About the Author—STAVROS J. PERANTONIS received his BS degree in Physics from the Department of Physics, University of Athens, an M.Sc. degree in Computer Science from the Department of Computer Science, University of Liverpool and a D.Phil. Degree in Computational Physics from the Department of Physics, University of Oxford. Since 1992 he has been with the Institute of Informatics and Telecommunications, NCSR "Demokritos", where he currently holds the position of Senior Researcher and Head of the Computational Intelligence Laboratory. His main research interests are in Image Processing and Document Image Analysis, OCR and Pattern Recognition.