Automatic Unsupervised Parameter Selection for Character Segmentation

G.Vamvakas, 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 Agia Paraskevi, Athens, Greece
{gbam, nstam, bgat, sper }@iit.demokritos.gr

ABSTRACT
A major difficulty for designing a document image segmentation methodology is the proper value selection for all involved parameters. This is usually done after experimentations or after involving a training supervised phase which is a tedious process since the corresponding segmentation ground truth has to be created. In this paper, we propose a novel automatic unsupervised parameter selection methodology that can be applied to the character segmentation problem. It is based on clustering of the entities obtained as a result of the segmentation for different values of the parameters involved in the segmentation method. The clustering is performed using features extracted from the segmented entities based on zones and from the area that is formed from the projections of the upper/lower and left/right profiles. Optimization of an appropriate intra-class distance measure yields the optimal parameter vector. The method is evaluated on two segmentation algorithms, namely a recently proposed character segmentation technique based on skeleton segmentation paths, as well as the well known RLSA technique. The proposed parameter selection method is capable of finding the segmentation parameters that correspond to the optimal or near optimal segmentation result, as this is determined by counting the number of matches between the entities detected by the segmentation algorithm and the entities in the ground truth.

Categories and Subject Descriptors
I.5.3 [**PATTERN RECOGNITION**]: Clustering – *algorithms, similarity measures*; I.4.6. [**IMAGE PROCESSING AND COMPUTING VISION**]: Segmentation; I.7.5. [**DOCUMENT AND TEXT DETECTION**]: Document Capture – *document analysis*.

General Terms
Algorithms, Performance, Design, Experimentation.

Keywords
Clustering, character segmentation, parameter selection.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
DAS '10, June 9-11, 2010, Boston, MA, USA
Copyright © 2010 ACM 978-1-60558-773-8/10/06... $10.00

1. INTRODUCTION
Character segmentation is a difficult problem since low quality of document images and the wide variety of fonts can cause touching and broken characters. Over the last decade, many different techniques have been proposed for character segmentation that can be classified into two main categories: (i) external character segmentation techniques that detect characters prior to recognition and (ii) internal character segmentation techniques that perform segmentation and recognition concurrently. Furthermore, in terms of methodology, character segmentation techniques can be categorized based on the document image segmentation algorithm that they adopt. The most known of these segmentation algorithms are the following: projection analysis, connected component analysis, Run Length Smoothing Algorithm (RLSA), contour shape analysis and Hough transform. Representative examples of character segmentation methodologies are the following: Antonacopoulos and Karatzas [1] use the horizontal projection profile of each word segment for character segmentation in historical machine-printed documents. This approach cannot handle the case of overlapping characters. Liang et al. [8] propose a character segmentation technique which is applied on modern machine-printed documents. Here, the touching characters problem is confronted using discrimination functions based on pixel and component profile projections. In [10], Nikolaou et al. use skeleton segmentation paths in order to isolate possible connected characters. The basic idea is to find all the possible segmentation paths linking the feature points on the skeleton of the word and its background. Then, using several criteria the best segmentation paths are selected. Xiao and Leedham [14] present a cursive script character segmentation method in which knowledge of the structure of English letters is investigated. First, connected components consisting of more than one character are split into sub-components based on their face-up or face-down background regions. Secondly, the over-segmented sub-components are merged into characters according to the knowledge of character structures and their joining characteristics. Finally, Chang and Chen [4] use the convex-hull techniques and typographical structure to segment touching characters without the guidance of recognition. Features based upon a convex hull are insensitive to character fonts and sizes, the touching-character problem of various fonts and sizes can be handled even for heavily touching characters or italic-type overlapping characters without slant correction. Table 1 summarizes the characteristics of those approaches [1, 8, 10, 14, 4] mentioned above.

In most segmentation approaches a major problem is the selection of the free parameters that affect directly the segmentation results (see Table 1). The parameters are either user-specified and no training method is included [1, 8, 10, 14, 4] or selected through a training procedure over a set of "optimal" parameter values that are usually manually selected based on some assumption regarding the training data [11], [6]. In general, automatic selection of the free parameters is actually an optimization problem [2]. Kanungo et al [9] propose an automatic methodology for the selection of the free parameters based on training. Mutually exclusive training and testing data sets are created with ground truth and an optimization procedure is adopted to search automatically for the optimal parameter values of the segmentation algorithm.

However, ground truth or a priori knowledge of the fonts of the document image is not always available. Therefore, there is need for unsupervised methods that can determine the optimal parameter values for character segmentation, without resorting to ground truth data. To this end, in this paper we propose a novel automatic unsupervised parameter selection methodology for character segmentation that is based on clustering. The clustering is performed using features extracted from the segmented entities based on zones and from the area that is formed from the projections of the upper/lower and left/right profiles. Optimization of an appropriate intra-class distance measure yields the optimal parameter vector. The remaining of the paper is organized as follows. In Section 2 the proposed methodology is introduced and the segmentation techniques used are presented in Section 3. Experimental results are shown in Section 4 and conclusions are drawn and discussed in Section 5.

Table 1: Representative character segmentation techniques and the parameters involved.

Character Segmentation Technique	Segmentation Algorithm	Parameters
Antonacopoulos and Karatzas [1]	Projection Profiles Analysis	A parameter which denotes the expected character box width defines the character separators
Liang et al. [8]	Projection Profiles Analysis	Segmentation discrimination functions are based on two parameters which denote the distances between adjacent columns
Nikolaou et al. [10]	Skeleton Segmentation Paths	Two main parameters define the area in which the algorithm searches for the best segmentation path
Xiao and Leedham [14]	Connected Component Analysis	Two main parameters confirm the merging of two sub-components
Chang and Chen [4]	Convex Hull	Four average widths of single characters are used to decide when to perform touching character detection

2. PROPOSED METHODOLOGY

A scheme relying on clustering is introduced for the selection of the segmentation parameters. The motivation for our approach is that parameters resulting in more compact clusters are more likely to produce a better segmentation result. Different segmentation results, from a set of document images, based on various vectors of parameter values are fed to the clustering algorithm. Then each clustering outcome is evaluated and the parameter vector that leads to the best clustering according to a specific criterion is considered to be the optimal vector for the segmentation procedure. Figure 1 shows the block diagram of the proposed methodology.

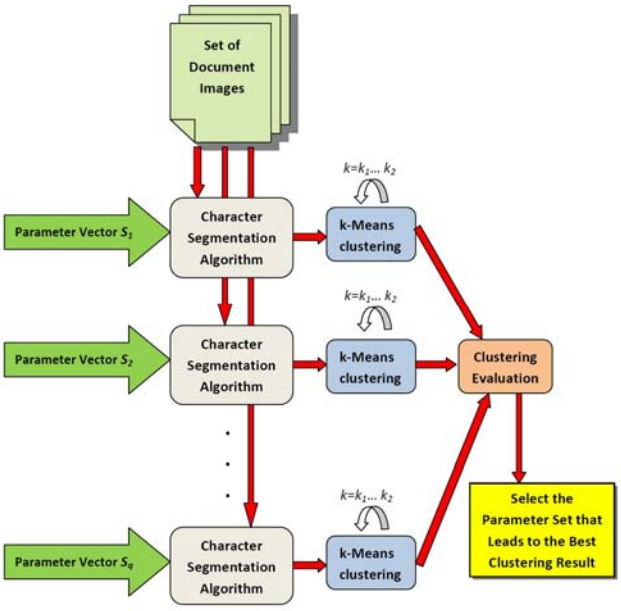
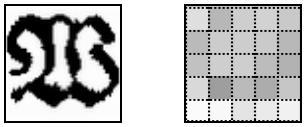


Figure 1: Block diagram of the proposed methodology.

2.1 Feature Extraction

Before the clustering technique is applied a feature extraction approach needs to be employed for all possible characters extracted at the segmentation stage. Firstly, all binary character images are normalized to an *H*x*H* matrix respecting the original aspect ratio. In our case *H* is set to 60.

We are based on features presented in [13] that have been proved efficient even for the difficult case of handwritten character recognition. The first set of features is based on zones. The image is divided into horizontal and vertical zones, and for each zone we calculate the density of the character pixels (see Figure 2).

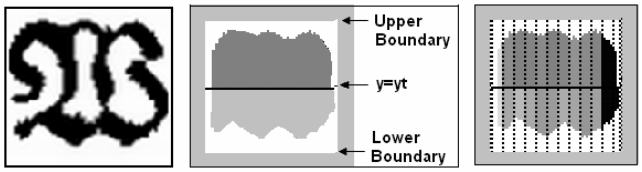


(a) (b)

Figure 2. Feature extraction of a character image based on zones; (a) The normalized character image ;(b) Features based on zones. Darker squares indicate higher density of character pixels.

In the second type of features, the area that is formed from the projections of the upper and lower as well as of the left and right character profiles is calculated. Firstly, the center mass (x_t, y_t) of the character image is found.

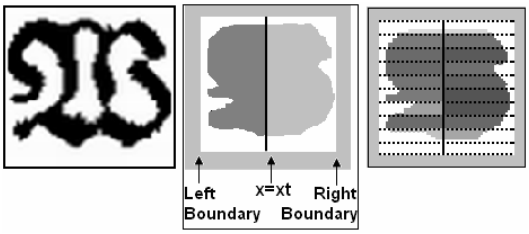
Upper/lower profiles are computed by considering, for each image column, the distance between the horizontal line $y=y_t$ and the closest pixel to the upper/lower boundary of the character image (see Figure 3b). This ends up in two zones (upper, lower) depending on y_t. Then both zones are divided into vertical blocks. For all blocks formed the area of the upper/lower character profiles is calculated. Figure 3c illustrates the features extracted from a character image using upper/lower character profiles.



(a) (b) (c)

Figure 3: Feature extraction of a character image based on upper and lower character profile projections; (a) The normalized character image; (b) Upper and lower character profiles; (c) The extracted features. Darker squares indicate higher density of zone pixels.

Similarly, the features based on left/right character profiles are extracted.



(a) (b) (c)

Figure 4: Feature extraction of a character image based on left and right character profile projections. (a) The normalized character image. (b) Left and right character profiles. (c) The extracted features. Darker squares indicate higher density of zone pixels.

In case of features based on zones the character image is divided into 5 horizontal and 5 vertical zones, thus resulting to 25 features. In case of features based on character (upper/lower) projection profiles the image is divided into 10 vertical zones, therefore the number of features is 20. Similarly, the image is divided into 10 horizontal zones so the number of features corresponding to features based on left/right projection profiles is also 20. Combination of features based on zones and features based on character profile projections led to the feature extraction model that uses a total of 65 features. So, every character is represented by a 65-dimensional feature vector F_i where i =1…65.

2.2 Parameter Selection based on Clustering

Consider a character segmentation algorithm whose result depends on P parameters. Let S_1, S_2 … S_v different parameter vectors (p-tuples) for different values of the parameters obtained using a standard selection method (e.g. random selection, selection through a grid).

In our approach the well-known k-Means clustering algorithm [12] is adopted due to its computational simplicity and the fact that, as all clustering techniques which use point representatives, is suitable for recovering compact clusters.

If the expected number of different characters is in the interval between k_1 and k_2, then for every S_q we proceed to a k-Means clustering with k taking values from k_1 to k_2. The steps of the k-Means algorithm are as follows:

Step 1: Initially choose the number of clusters to be k.

Step 2: Choose arbitrary a set of k instances as initial centres of the clusters.

Step 3: Each instance is assigned to the cluster which is closest.

Step 4: The cluster centroids are recalculated either after each instance assignment.

Step 5: *GOTO* Step 3 until no change occurs between two successive iterations.

Given a parameter vector S_q, in order to evaluate the performance of the clustering algorithm for every k between k_1 and k_2, the mean squared distances from the centroids (within clusters sum of squares) is calculated as follows:

$$W_q(k) = \sum_{j=1,2..k} \frac{1}{n_{c_j}} \sum_{i \in C_j} d^2(x_i, \overline{x_j}) \qquad (1)$$

where $\overline{x_j}$ is the centroid of the cluster C_j, j = 1,2 … k , x_i is the i_{th} pattern inside cluster C_j , n_{c_j} is the cardinality of cluster C_j and d is the Euclidean Distance.

The value of $W_q(k)$ is low when the partition is good thus resulting to compact clusters. Table 2 presents an example of the computed $W_q(k)$ using an RLSA based segmentation approach for various parameter vectors S_q and k values. The optimal parameter vector, in this example, that minimizes $W_q(k)$ is found to be S_3 when k = 60.

Table 2: Example of $W_q(k)$ values.

	$W_q(k)$		
	k = 50	**k = 60**	**k =70**
q=1	4044577	3699033	2568287
q=2	3544091	2955565	2433272
q=3	3437220	**2115794**	2744399
q=4	4317143	3054617	2903489
q=5	3414985	2766293	2581995
q=6	3238709	2632454	2427183

A measure of the quality of the segmentation result that corresponds to a parameter vector S_q is given as:

$$Q(S_q) = \frac{10^5}{\min_{k=k_1,...,k_2}(W_q(k))} \qquad (2)$$

The optimal parameter vector S_{opt} is defined as:

$$S_{opt} = \arg\max_{S_q = S_1, S_2, ..., S_v} (Q(S_q)) \qquad (3)$$

3. Character Segmentation Algorithms and their Parameters

In order to illustrate the performance of the proposed methodology two character segmentation approaches were used: a) the one presented in [10] that use skeleton segmentation paths and b) the well-known RLSA [7].

3.1 Using Skeleton Segmentation Paths

In [10] the basic idea is to find all the possible segmentation paths linking the feature points on the skeleton of the word and its background. Then, several criteria are used in order to select the best segmentation paths. First, the dominant character height (*LettH*) is calculated. Then, the connected components (*CCs*) of a word are calculated and finally, all the *CCs* that have their height to width ratio less or equal to 0.5 are examined in order to separate them into characters. During this procedure, we take into consideration only segmentation paths that result to characters with width in the limit of [*MinCharWidth*LettH*, *MaxCharWidth*LettH*]. The two main parameters of the character segmentation technique are the *MinCharWidth* and *MaxCharWidth*. These parameters define the area in which the algorithm searches for the best segmentation path. In Figure 5 segmentation results using different parameter vectors are presented.

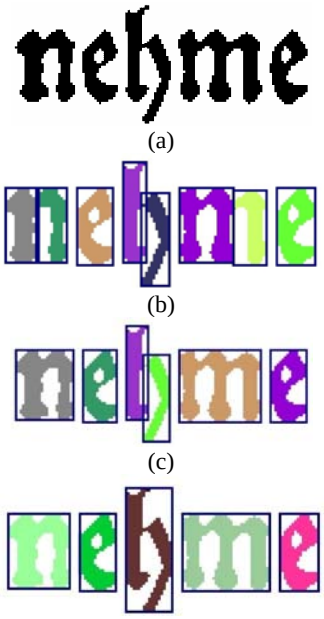


Figure 5: Example of character segmentation using the method presented in [10]; (a) original image; Segmentation results using for the *MinCharWidth* and *MaxCharWidth* parameters the following pairs: (b) (0.4, 0.8); (c) (0.5, 1.3); and (d) (0.7, 0.9).

3.2 Using RLSA

The RLSA based approach for character segmentation examines the white runs existing in the vertical direction and then the white runs with length less than a threshold *Th* are eliminated [7] (see Figure 6). This threshold depends on the calculated dominant character height *LettH* and is defined as *a* LettH* where *a* is the main parameter of the algorithm.

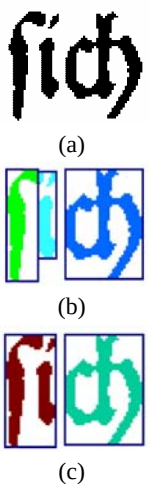
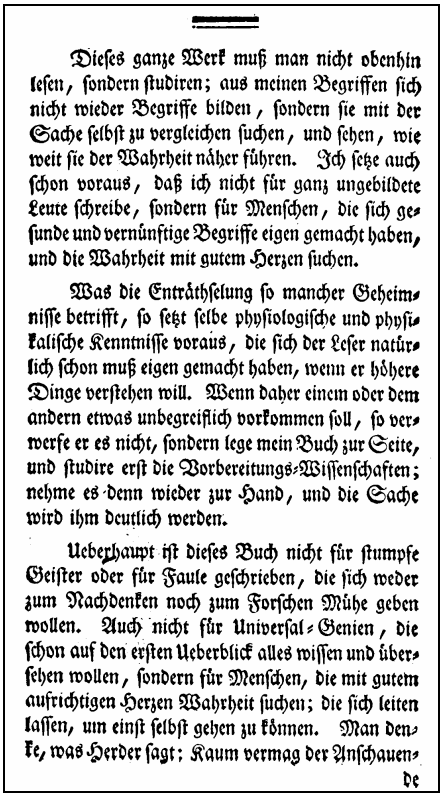


Figure 6: Example of character segmentation using the RLSA method; (a) original image; Segmentation results using for the *a* parameter the following values: (b) 0.2; and (c) 0.7.

4. EXPERIMENTAL RESULTS

The proposed methodology was tested on a part of a historical book from Eckartshausen which was published on 1788 and is owned by the Bavarian State Library [3] consisting of 20 document images that contained 16398 characters (see Figure 7).



Dieſes ganze Werk muß man nicht obenhin leſen, ſondern ſtudiren; aus meinen Begriffen ſich nicht wieder Begriffe bilden, ſondern ſie mit der Sache ſelbſt zu vergleichen ſuchen, und ſehen, wie weit ſie der Wahrheit näher führen. Ich ſetze auch ſchon voraus, daß ich nicht für ganz ungebildete Leute ſchreibe, ſondern für Menſchen, die ſich geſunde und vernünftige Begriffe eigen gemacht haben, und die Wahrheit mit gutem Herzen ſuchen.

Was die Enträthſelung ſo mancher Geheimniſſe betrifft, ſo ſetzt ſelbe phyſiologiſche und phyſikaliſche Kenntniſſe voraus, die ſich der Leſer natürlich ſchon muß eigen gemacht haben, wenn er höhere Dinge verſtehen will. Wenn daher einem oder dem andern etwas unbegreiflich vorkommen ſoll, ſo verwerfe er es nicht, ſondern lege mein Buch zur Seite, und ſtudire erſt die Vorbereitungs-Wiſſenſchaften; nehme es denn wieder zur Hand, und die Sache wird ihm deutlich werden.

Ueberhaupt iſt dieſes Buch nicht für ſtumpfe Geiſter oder für Faule geſchrieben, die ſich weder zum Nachdenken noch zum Forſchen Mühe geben wollen. Auch nicht für Univerſal-Genien, die ſchon auf den erſten Ueberblick alles wiſſen und überſehen wollen, ſondern für Menſchen, die mit gutem aufrichtigen Herzen Wahrheit ſuchen; die ſich leiten laſſen, um einſt ſelbſt gehen zu können. Man denke, was Herder ſagt; Kaum vermag der Anſchauende

Figure 7: Sample of the document images used.

4.1 Performance of the Proposed Method

For the *k*-Means clustering algorithm, *k* takes values between k_1 = 35 and k_2 = 80. These values are considered to be suitable enough since most alphabets have more than 35 and less than 80 characters.

For the *MinCharWidth* and the *MaxCharWidth* parameters in [10] the following pairs were used: S_1=(0.4, 0.6), S_2=(0.4, 0.8), S_3=(0.4, 1.0), S_4=(0.4, 1.2), S_5=(0.4, 1.4), S_6=(0.5, 0.7), S_7=(0.5, 0.9), S_8=(0.5, 1.1), S_9=(0.5, 1.3), S_{10}=(0.5, 1.5), S_{11}=(0.6, 0.8), S_{12}=(0.6, 1.0), S_{13}=(0.6, 1.2), S_{14}=(0.6, 1.4), S_{15}=(0.7, 0.9), S_{16}=(0.7, 1.1), S_{17}=(0.7, 1.3), S_{18}=(0.7, 1.5). For each one the procedure described in Section 2.2 is performed and as shown in Figure 8 the optimal pair is S_{opt}=S_{15}=(0.7, 0.9).

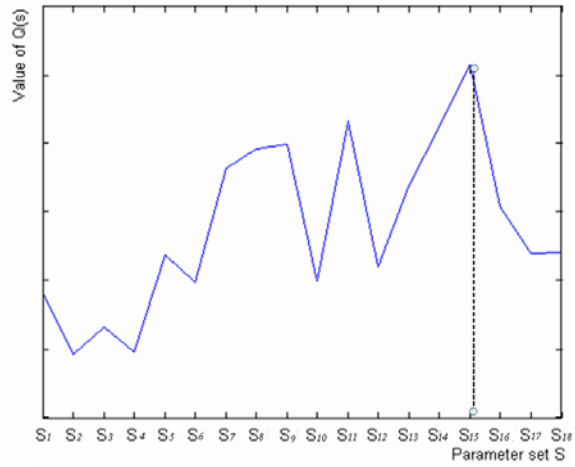
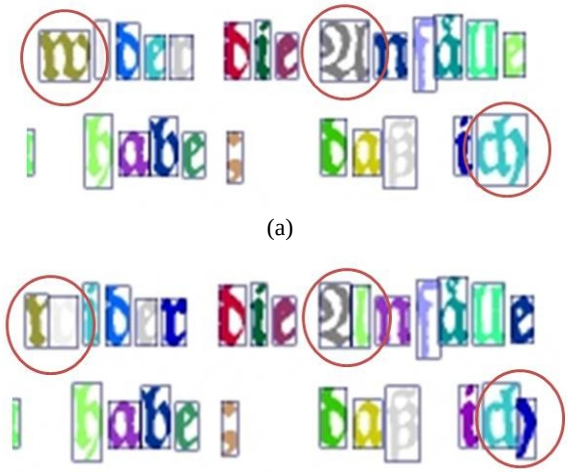


Figure 8: Optimal pair of parameters *MinCharWidth* and *MaxCharWidth*.

Figure 9 illustrates an example of segmentation results for S_2 and S_{opt}=S_{15} respectively. It is evident that when the parameters of S_{15} are used the segmentation algorithm performs avoids over segmentation errors.



(a)

(b)

Figure 9: Example of character segmentation using the method presented in [10] using as parameters (a) the optimal pair S_{opt}= S_{15}= (0.7, 0.9) and (b) S_2= (0.4, 0.8).

The *a* parameter for the RLSA based segmentation was search in the range of [0.2, 0.7] and the optimal value was found to be a = 0.4 (see Figure 10). In Figure 11 one can observe that for a value *a* = 0.7 different form the optimal one the segmentation procedure fails since it results to merging characters from different text lines.

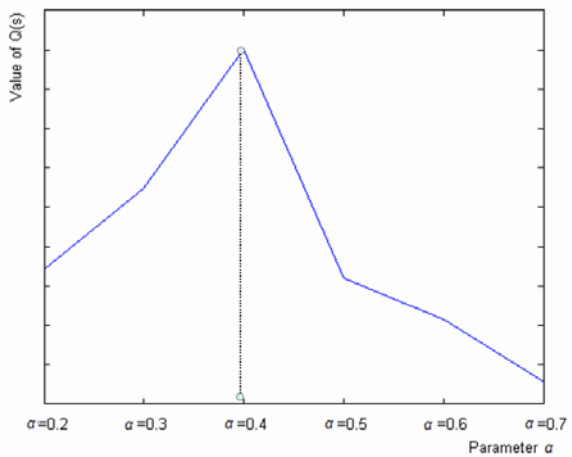
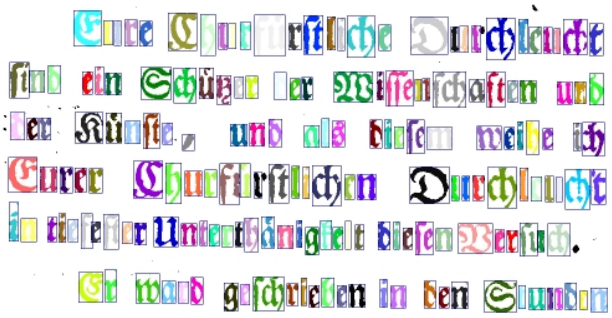
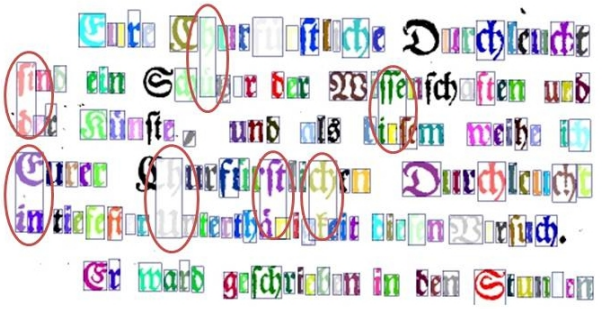


Figure 10: Optimal value of parameter *a* for RLSA based Segmentation.



(a)



(b)

Figure 11: Example of character segmentation using the RLSA method using as parameter (a) the optimal value *a* = 0.4 (optimal) and (b) *a* = 0.7.

4.2 Evaluation

In order to have quantitative evaluation of the proposed methodology, we manually marked the correct character segments in the dataset and used an automatic evaluation procedure. The

performance evaluation method used was based on counting the number of matches between the entities detected by the segmentation algorithm and the entities in the ground truth [5]. Let I be the set of all image 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:

$$MatchScore(i,j) = \frac{T(G_j \cap R_i \cap I)}{T((G_j \cup R_i) \cap I)} \quad (4)$$

We consider a region pair as a one-to-one match only if the matching score is equal to or above the evaluator's acceptance threshold T_a. If N is the count of ground-truth elements, M is the count of result elements, and $o2o$ is the number of one-to-one matches, we calculate the detection rate (DR) and recognition accuracy (RA) as follows:

$$DR = \frac{o2o}{N} \quad , \quad RA = \frac{o2o}{M} \quad (5)$$

A performance metric FM can be extracted if we combine the values of detection rate and recognition accuracy:

$$FM = \frac{2\ DR\ RA}{DR + RA} \quad (6)$$

Tables 3 and 4 depict the results for the segmentation method presented in [10] and for the RLSA based segmentation where the acceptance threshold is set to $T_a = 90$.

Table 3: Evaluation of Character Segmentation Method [10].

Parameter Vector S_q **=(*MinCharWidth, MaxCharWidth*)**	**FM**
S_1 = (0.4, 0.6)	72.3%
S_2 = (0.4, 0.8)	71.9%
S_3 = (0.4, 1.0)	72.1%
S_4 = (0.4, 1.2)	72.2%
S_5 = (0.4, 1.4)	72.3%
S_6 = (0.5, 0.7)	80.7%
S_7 = (0.5, 0.9)	80.1%
S_8 = (0.5, 1.1)	80.1%
S_9 = (0.5, 1.3)	80.1%
S_{10} = (0.5, 1.5)	80.1%
S_{11} = (0.6, 0.8)	82.9%
S_{12} = (0.6, 1.0)	82.7%
S_{13} = (0.6, 1.2)	82.6%
S_{14} = (0.6, 1.4)	82.6%
S_{15} = (0.7, 0.9)	**83.8%**
S_{16}= (0.7, 1.1)	83.6%
S_{17} = (0.7, 1.3)	83.6%
S_{18} = (0.7, 1.5)	83.6%

From Table 3 is evident that the best segmentation result is achieved when *MinCharWidth*= 0.7 and *MaxCharWidth* = 0.9 that is the pair found by the proposed methodology (see Section 3.2).

Table 4: Evaluation of RLSA Character Segmentation.

Parameter *a*	**FM**
α = 0.2	81.6%
α = 0.3	**82.3%**
α = 0.4	81.4%
α = 0.5	79.8%
α = 0.6	79.3%
α = 0.7	78.5%

As shown in Table 4 although the optimal value for parameter a should be 0.3 the proposed methodology suggested that it should be 0.4. However, the performance evaluation for these two values is relatively close.

5. CONCLUSIONS

In this paper we propose a methodology for automatic unsupervised parameter selection for character segmentation. The methodology is based on clustering; suggesting that the optimal segmentation output, relying on a vector of parameters, should produce the best clustering. Different vectors of parameters are used over two segmentation techniques thus resulting to different clustering outcomes and then, every clustering is evaluated using an appropriate intra-class distance measure. For each segmentation approach, the parameter vector that led to the best clustering is selected to be the optimal vector for the segmentation algorithm. Experimental results, based on evaluation of segmentation using the ground truth, show that the proposed methodology is capable of finding the optimal or near optimal parameter vector. Our future work will focus on combining techniques or exploiting new ones in order to improve the current performance as well as eliminating, to the point that this is reachable, the cost for selecting the optimal parameter vector for segmentation.

6. ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement N° 215064 (project IMPACT).

7. REFERENCES

[1] Antonacopoulos, A., Karatzas, D., 2005. Semantics-based content extraction in typewritten historical documents, in: Eighth International Conference on Document Analysis and Recognition, 48–53.

[2] Bishop C.M., 1995, Parameter Optimization Algorithms, Neural Networks for Pattern Recognition: Oxford University Press, pp. 253-292.

[3] Carl von Eckartshausen, 1778, Aufschlüsse zur Magie aus geprüften Erfahrungen über verborgene philosophische Wissenschaften und verdeckte Geheimnisse der Natur", Bavarian State Library.

[4] Chang, T.C., Chen, S.Y., 1999, Character Segmentation Using Convex-Hull Techniques, International journal of pattern recognition and artificial intelligence, 833-858.

[5] Gatos G., Stamatopoulos N. and Louloudis G., 2009, ICDAR2009 Handwriting Segmentation Contest, 10th International Conference on Document Analysis and Recognition, pp. 1393-1397.

[6] Kavallieratou E., Stamatatos E., Fakotakis N., Kokkinakis G., 2000, Handwritten character segmentation using transformation-based learning, 15th International Conference on Pattern Recognition, vol. 2, pp. 634–637.

[7] Konidaris T., Gatos B., Ntzios K., Pratikakis I., Theodoridis S. and Perantonis S.J., 2007, Keyword - Guided Word Spotting in Historical Printed Documents Using Synthetic Data and User feedback , International Journal on Document Analysis and Recognition (IJDAR), special issue on historical documents, Vol. 9, No. 2-4, pp. 167-177, 2007.

[8] Liang, S., Shridhar, M., and Ahmadi, M., 1994. Segmentation of touching characters in printed document recognition, Pattern Recognition 27 (6), 825–840.

[9] Mao S. and Kanungo T., 2001, Empirical Performance Evaluation Methodology and Its Application to Page Segmentation Algorithms, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, pp. 242-256.

[10] Nikolaou, N., Makridis, M., Gatos, B., Stamatopoulos N., and Papamarkos, N., 2009. Segmentation of historical machine-printed documents using Adaptive Run Length Smoothing and skeleton segmentation paths", Image and Vision Computing, doi:10.1016/j.imavis.2009.09.013.

[11] Philips T. and Chhabra A.K., 1999, Empirical Performance Evaluation of Graphics Recognition Systems, IEEE Transactions on Pattern Analysis and Machine Intelligence, 21:849-879.

[12] Theodoridis, S., and Koutroumbas, K., 1997, Pattern Recognition, Academic Press.

[13] Vamvakas G., Gatos B., Petridis S. and Stamatopoulos N., 2007, An Efficient Feature Extraction and Dimensionality Reduction Scheme for Isolated Greek Handwritten Character Recognition, 9th International Conference on Document Analysis and Recognition (ICDAR'07), pp. 1073-1077.

[14] Xiao, X., and Leedham, G., 2000. Knowledge-based English cursive script segmentation, Pattern Recognition Letters 21 (10), 945–954.