Comparative Study of Text Line Segmentation Algorithms on Low Quality Documents

Pulagam Soujanya[1], Vijaya Kumar Koppula[2], Kishore Gaddam[3] & P. Sruthi[4]

[1,2,4] CMR College of Engineering and Technology
[3]Cognizent Technologies, Hyderabad, India
E-mail : Soujanya.pulagam@gmail.com, vijaykoppula@gmail.com, Kishore.gr@gmail.com

Abstract **-** Segmentation of text lines is one of the important steps in the Optical Character Recognition system. Text Line Segmentation is pre-processing step of word and character segmentation. Text Line Segmentation can be viewed simple for printing documents which contains distinct spaces between the lines. And it is more complex for the documents where text lines are overlap, touch, curvilinear and variation of space between text lines like in Telugu scripts and skewed documents. The main objective of this project is to investigate different text line segmentation algorithms like Projection Profiles, Run length smearing and Adaptive Run length smearing on low quality documents. These methods are experimented and compare their accuracy and results.

Key words *- segmentation, handwriting, text lines, Historical documents, degraded documents.*

I. INTRODUCTION

Optical Character Recognition is presently one of the important enabling technologies for the progress of language oriented work. The transformation of paper media text into the searchable and computer revisable format gives research in the field of language technologies a great boost. OCR is being used in the context of language technology research for creation of text corpora. Office automation and content creation activities are the important Areas of OCR applications.

OCR system for printed good quality documents of European languages has addressed by the researchers. However printed Telugu script segmentation is a challenging problem due to the complex orthography with a large number of distinct character shapes, estimated to be around 10,000, composed of simple and compound characters formed from 16 vowels, called acchulu, and 36 consonants, called hallulu. In addition several semi-vowel symbols, called maatra, are used in conjuction with hallus and half consonants, called voththulu are used in consonant clusters In this script we may not find consonant space between two lines due to consonant modifier and vowel modifiers. Most of the Historical machine-printed documents often suffer from low quality and several degradations due to the old printing matrix quality or ink diffusion, background noise, artifacts due to aging, primary typeset and interfering lines. Because of these reasons the documents may contains:

- Curvilinear text lines
- Neighbouring text lines may be close or touch each other
- Variations in font sizes
- Non-constant spaces between text lines.
- Skewed text

Fig. 1: Shows sample documents of overlap, touch and skewed.

The objective of this project is to investigate line segmentation approaches like Projection profile, Run length smearing, Adaptive Run Length smearing and applying these algorithms on low quality documents.

This paper is organized as follows: Section 2 describes Investigation of Line Segmentation Methods, section 3 we discuss the Evaluation Methodology and section 4 describes the conclusions of this paper.

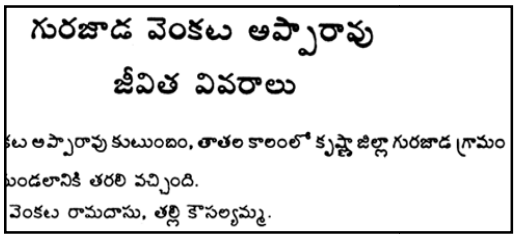


Fig (a)

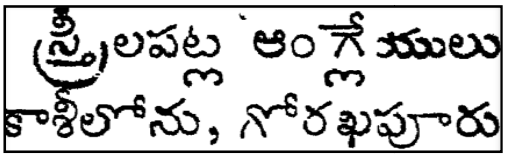


Fig (b)

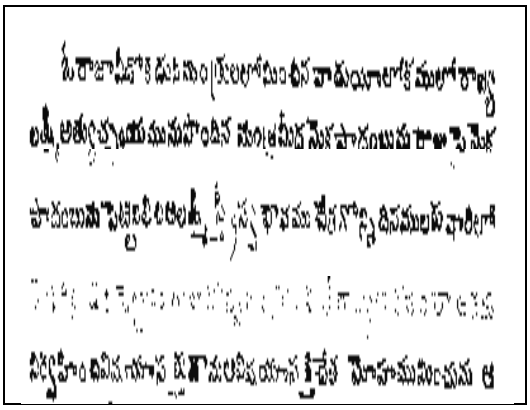


Fig (c)

Fig 1. Example Image for a) Various font sizes b) Neighbouring lines overlap or touch each other c) Skewed document

II. LINE SEGMENTATION METHODS INVESTIGATED

2.1. Projection Profile Based Algorithm

Projection-profiles are commonly used for printed document segmentation. This technique can also be adapted to handwritten documents with little overlap[2]. The vertical projection profile is obtained by summing pixel values along the horizontal axis for each y value. From the vertical profile, the gaps between the text lines in the vertical direction can be observed (Fig. 2). The vertical profile is not sensitive to writing fragmentation. Variants for obtaining a profile curve may consist in projecting black/white transitions such as in Marti and Bunke [9] or the number of connected components, rather than pixels. The profile curve can be smoothed, e.g. by a Gaussian or median filter to eliminate local maxima [10]. The profile curve is then analyzed to find its maxima and minima. There are two drawbacks: short lines will provide low peaks, and very narrow lines, as well as those including many overlapping components will not produce significant peaks. In case of skew or moderate fluctuations of the text lines, the image may be divided into vertical strips and profiles sought inside each strip (Zahour *et al.* [11]). These piecewise projections are thus a means of adapting to local fluctuations within a more global scheme. In Shapiro *et al.*[12], the global orientation (skew angle) of a handwritten page is first searched by applying a Hough transform on the entire image. Once this skew angle is obtained, projections are achieved along this angle. The number of maxima of the profile gives the number of lines. Low maxima are discarded on their value, which is compared to the highest maxima. Lines are delimited by strips, searching for the minima of projection profiles around each maximum. In the work of Antonacopoulos and Karatzas [13], 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. The method is applied to printed records of the second World War which have regularly spaced text lines. The logical structure is used to derive the text regions where the names of interest can be found

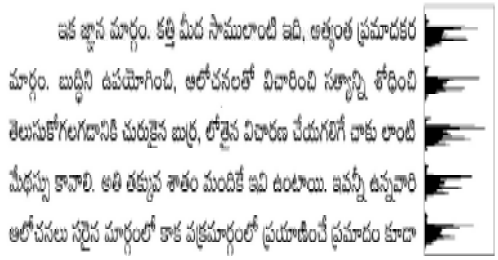


Fig: 2 : Binarized image of the image and it's corresponding Horizontal Projection Profile.

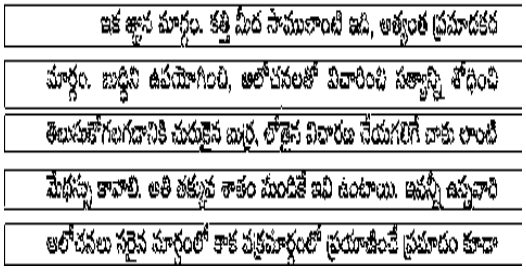


Fig: 3 :Line Segmented output.

2.2. Run Length Smearing Method

For printed and binarized documents, smearing methods such as the Run-Length Smoothing Algorithm can be applied. Consecutive black pixels along the horizontal direction are smeared: i.e. the white space between them is filled with black pixels if their distance is within a predefined threshold. The bounding boxes of the connected components in the smeared image enclose text lines .In this step we use run length smearing technique to increase the strength of the histogram. Here we consider the consecutive run of white pixels in between two black pixels and we compute the length of that white run. If the length of white run is less than five times of stoke width (thickness of a line in a font character), we fill up the white run length into black[5]. In Fig 4 (a) there are two original text lines and in Fig 3.2.1(b) there are smoothed text lines with horizontal histogram corresponding of their two text lines.

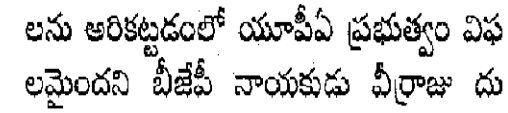


Fig:4 (a) : Original text lines

Fig. 4(b) : Smoothed text lines with histogram

2.3. Adaptive Run Length Smearing Algorithm

2.3.1 Method Used For Line Segmentation

In this method, combine ideas from connected component processing, whitespace analysis and skeleton representation and introduce several innovative aspects in order to achieve a successful segmentation of historical machine-printed documents. This technique does not require any parameter tuning by the user and can deal with degradations which occur due to shadows, non-uniform illumination, low contrast, large signal-dependent noise, smear and strain. The average character height AH for the document image is then calculated in order to be used for parameter tuning and font size independency in all processes[5]. The b/w image is first DE skewed and pre-processed by removing noisy black borders and noisy text areas appearing from neighbouring pages as well as small noisy components which can have a negative effect on the statistical measures of the document image. Punctuation marks are initially eliminated and combined with a later stage result of the method in order for the text line segmentation stage to be completed more accurately. The resulted image of this stage is smeared with a proposed Adaptive Run Length Smoothing Algorithm (ARLSA) that helps grouping together homogeneous text regions. At the next stage, obstacles are detected and used in order to isolate different text lines and different text columns.

Obstacles are considered as regions within a document that a horizontal run length procedure is not allowed to cross. Using obstacles, the initial text line segmentation result is efficiently calculated. By combining this result with the punctuation marks, the final text line segmentation is performed. In the next stage, each detected text line is processed independently to extract the word segments. Based on the histogram of horizontal distances between Adjacent bounding boxes, a proper threshold value is calculated and used in order to merge components belonging to the same word. Finally, character segments are extracted from each word segment based on skeleton segmentation paths which are used to isolate possible connected characters.

2.3.2 Adaptive Run Length Smoothing Method

In the segmentation technique, a modified version of the horizontal RLSA is proposed, the Adaptive Run Length Smoothing Algorithm (ARLSA), in order to overcome the drawbacks of the original algorithm, such as grouping inhomogeneous components or different slanted lines. ARLSA also works successfully with documents containing characters with variable font size. Before the application of ARLSA a connected component analysis is necessary. Two types of background pixels (white) sequences are considered. The first type concerns sequences which occur between two foreground pixels (black) which belong to the same connected component as shown in Fig. 5(a) In this case, all background pixels of the sequence are replaced with foreground pixels. The second type of background pixels sequence occurs between two different connected components Fig. 5(b).In this case, constraints are set in regard to the geometrical properties of the connected components and the replacement with foreground pixels is performed when these constraints are satisfied.

Fig. 5(a) : Sequences which occur between two foreground pixels (black) which belong to the same connected component

Fig. 5(b) : Sequence which occur between two different Connected components

Let CC_i and CC_j be two connected components and S(i,j) a horizontal sequence of background pixels between CC_i and CC_j ..

We define the following four metrics:

1. L(S): length of the sequence S(i,j),that is the number of white pixels.
2. $H_R(s)$: height ratio between CC_i and CC_j, which is defined as follows:

$$H_R(s) = \frac{\text{Max}\{h_i,h_j\}}{\text{Min}\{h_i,h_j\}}$$

3. $O_H(s)$: the horizontal overlapping between the bounding boxes of CC_i and CC_j , which is defined by the following equation:

$$O_H(s)=\max\{Yl_i ,Yl_j\} - \min\{Yr_i ,Yr_j\}$$

Where $\{Xl_i ,Yl_j\}$ $\{Xr_i ,Yr_j\}$the coordinates of the upper left and down right corner of the CCi's bounding box. The horizontal overlapping $O_H(s)$ is graphically demonstrated in Fig:6 and it can be observed that when $O_H(s) < 0$, horizontal overlapping exists between the two connected components CCi and CCj.

4. N(S): a binary output function.

N(S) is set to 0 when in the 3 * 3 neighbourhood of at least one pixel of the sequence S(i,j), a third connected component CC_K; k # i; j exists.Otherwise, N(S) is set to 1.

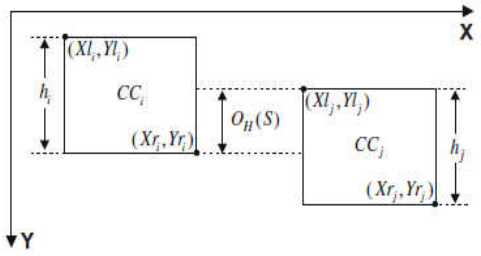


Fig. 6 : Graphical depiction of horizontal overlapping

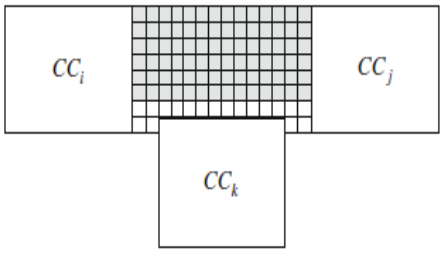


Fig. 7 : Graphical depiction of the constraint based on the N(S) function. Gray pixels represent the background pixels which will be replaced with foreground pixels

Based on the above metrics, a sequence of background pixels is replaced with foreground pixels only if:

$$(L(S)<=T_l)\wedge(H_R(s)<=T_h)\wedge(O_H(s)>=T_o)\wedge (N(S)=1 \quad (1)$$

Where $T_l, T_h,$To are predefined threshold values. The length threshold T_l of each sequence is related to the heights of the connected components. If h_i and h_j express the heights of CC_i and CC_j then

- $T_l = a.\min\{h_i,h_j\}$ where a is constant.
- Preferred value for a is 5.

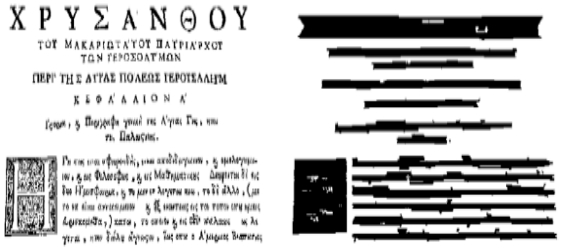
Where a is a constant value. Threshold T_h was set to 3.5 based on the following assumption: We consider that between a lowercase character such as ''o" and a character with descender or ascender of the same font size, a height ratio between 2 and 3 is very common. Taking into account the fact that in historical documents the font size varies we concluded to value 3.5 for T_h which gives enough tolerance against font size variation between characters of the same text line.

The horizontal overlapping threshold To is expressed as the percentage of overlap in regard to the component with the smallest height, that is:

- $To = c.\min\{h_i,h_j\}$

Where c is set to 0.4. This means that at least 40% of the shortest component height must be covered in order for a link to be established.

The constraint based on the function N(S), ensures that background pixels will be transformed into foreground pixels only if in their 3 * 3 neighbourhood no pixels of a third connected component exists as shown in Fig7 Its purpose is to prevent the creation of false links between objects and therefore the integration into component groups of unwanted objects. It is very helpful in historical and degraded documents where text line spacing is narrow and characters from different text lines overlap. In the example of Fig.7 the application of the ARLSA, without taking into account this constraint, would result in the creation of a group containing all three components CC_i, CC_j, CC_k. which is obviously wrong. The ARLSA in regard to the original algorithm can prevent the creation of inhomogeneous groups of components, namely to have large and small characters grouped together. Also, it has tolerance against warped or skewed text, that is components of different text lines that are close to each other at the horizontal direction is not likely to be linked even when T_l receives large values. This happens due to the horizontal overlapping constraint. For example in the case of Fig 8, in which the ARLSA is applied, all text elements were correctly formed into text lines without links with the graphic element. The value of factor a (Eq.1) used here is 5. This is a large value and it ensures that even distant objects will be joined if the other three conditions of the ARLSA procedure are satisfied.



(a) (b)

Fig.8 (a) : Original image, b) ARLSA (a =5; T_h = 3.5, c =0.4).

2.3.3 Noise and punctuation marks removal

The purpose of this stage is to remove small noisy connected components and to erase punctuation marks in order to improve the structure of the background and simplify the following procedures of the technique. First, noisy elements are filtered out based on three characteristics of the connected components and their corresponding bounding boxes. For a connected component CC_i these characteristics are:

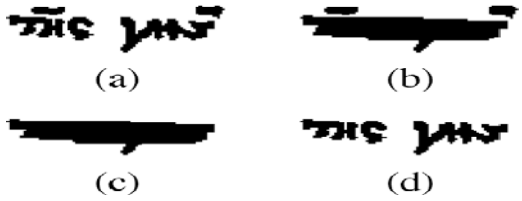
- The height of the bounding box of the CC_i; $H(CC_i)$.
- The elongation $E(CC_i) = \dfrac{\text{Min}\{H(cc_i),W(cc_i)\}}{\text{Max}\{H(cc_i),W(cc_i)\}}$

The measure shows the ratio of the shorter to the longer side of each bounding box.

- Density $D(cc_i) = \dfrac{P_{num}(cc_i)}{BB_{size}(cc_i)}$

which is the ratio of the number of foreground pixels P_{num} (cc_i) to the total number of pixels in the bounding box BB_{size} (cc_i) = min{$H(cc_i)$,$w(cc_i)$} Connected components with $H(CC_i) < AH/3$, or $D(CC_i) < 0.08$,or $E(CC_i) < 0.08$ are considered as noisy elements and they are eliminated. These values have been selected very carefully so no character elements will be eliminated. With this type of filtering only very noisy non-character objects are removed. The second type of filtering removes punctuation marks. It is based on the comparison of the connected components from two images, the initial document image and the resulted image after min{H(cci), w(cci)} the application of the ARLSA with a = 1.5. Let I_1 be the original image (see Fig.9a), I_2 the image after the application of the ARLSA (see Fig9b). The number of pixels PI_2 of each connected component $CC_i \sum I_2$ is calculated, that is the number of the black pixels. In the defined area of each $CC_i \sum I_2$, the sum PI_1 of the corresponding black pixels of I_1 is also calculated. The ratio of these two sums is taken into account as in the fallowing equation:

$$P_R = \frac{P_{L2}}{P_{L1}} \leq T_R$$



(a) (b) (c) (d)

Fig. 9 :Punctuation marks removal: (a) original image I_1, (b) application of ARLSA(I_2), (c) components with small pixel size ratio are removed (I_3) and (d) resulted image after the operation I_1 and I_3.

Components which correspond to pixel size ratio smaller than T_R are removed and a new image I_3 (see Fig.9c) is produced. Punctuation marks are removed because they are mainly isolated objects. This means that after the application of the ARLSA their size is likely to remain constant or change by a small factor contrary to text components. The final result is obtained by an AND operation between images I_1 and I_3 as shown in Fig.9. A proper value for T_R was found to be 1.15. This means that PI_2 is expected to be 15% larger than PI_1 .The punctuation marks eliminated in this stage are used in the text line segmentation stage in order to extract the final form of text lines.

2.3.4 Text line segmentation

Text line segmentation is performed by applying the ARSLA algorithm to the original image, after the noise and punctuation marks removal. Furthermore, ARLSA is constrained by text line and column obstacles. Therefore, ARLSA is performed only in cases where a background (white) pixel sequence does not include pixels that they have been also detected as obstacles. Constant a (see Eq (1)) is set to 5, which is a relative large value, so as distant parts of the same text line can be linked if the other three conditions of the ARLSA are satisfied. Obstacles prevent parts of different text lines to be linked despite of the large value of a. Finally, the punctuation marks that were removed, as described in Section2.3.3, are now combined with their nearest text line to complete the text line segmentation process

III. EVALUATION METHODOLOGY

Traditionally in machine printed document analysis, the text line segmentation results are represented as rectangular bounding boxes. Therefore, the evaluation is often based on the four coordinates of a bounding box.

However, some overlaps inevitably occur among bounding boxes in curvilinear handwritten documents. Non overlapping closed curves are better in the sense of representing curvilinear handwritten text lines. By this representation, the evaluation can be done at the pixel level, which is more accurate than that done at the bounding box level. Supposing there are M ground-truthed lines and N detected lines, we construct an M × N matrix P. An element $P_{i,j}$ for i=1, …,M and j = 1, . . .,N of matrix P represents the number of shared black pixels between the i^{th} ground-truthed text line and the j^{th} detected line[2]. We enforce one-to-one correspondence between the detected lines and the ground truth. Since the number of lines in two sets is different in general, we augment the matrix P to a square matrix P^1. A line is allowed to be matched to a dummy line and this match has no shared pixels ($P_{i,j}$ = 0). The square matrix P^1 has a dimension max (M,N) × max(M,N). For each assignment of the correspondence for ground truth S (k),k =1, . . , max (M,N), the goodness G(S) of this assignment is the total number of shared black pixels:

$$G(s) = \sum_{K=0}^{Max\ (M,N)}$$

The best assignment So is the one with maximum goodness: S_o=arg max G(S)

The Hungarian algorithm is used to efficiently search for the assignment problem. The overall pixel-level hit rate is defined as the best assignment So is the one with

$$H = \frac{G(S_0)}{\text{Number of black pixels in the Ground}}$$

By using the pixel-level hit rate and the Hungarian algorithm, different segmentation errors, for example, splitting, merging, and missing, can be appropriately penalized with weights that are proportional to the number of pixels involved. We can also evaluate the performance at the text line level. One ground-truthed line i is claimed to be correctly detected if

$$H = \frac{P_{i,so(i)}}{\sum_{j=1}^{N} P_{i,j}} \geq 0.9$$

And

$$H = \frac{P_{i,so(i)}}{\sum_{K=1}^{M} P_{i,so(i)}} \geq 0.9$$

In other words, if a ground-truthed line and the corresponding detected line share at least 90 percent of the pixels with respect to both of them, a text line is claimed to be correctly detected. In the following experiments, we evaluate the performance based on both the pixel-level hit rate and the text-line-level detection rate.

3.1 Results of 3 Algorithms.

DT is Detected Lines and GT is Ground Truth Lines

Telugu Documents																
Method Name	Overlap				Non Constant Spaces				Well Printed				Diff Font Sizes			
	GT	DT	DT 100%	DT 95%	GT	DT	DT 100%	DT 95%	GT	DT	DT 100%	DT 95%	GT	DT	DT 100%	DT 95%
Projection Profile	50	39	22	22	38	40	26	28	41	40	35	38	29	30	28	29
Run Length Smearing	50	56	0	0	38	44	18	26	41	36	15	17	29	44	16	22
Adaptive Run Length Smearing	50	75	0	0	38	79	0	0	41	62	0	0	29	44	15	29

English Documents																
Method Name	Overlap				Non Constant Spaces				Well Printed				Diff Font Sizes			
	GT	DT	DT 100%	DT 95%	GT	DT	DT 100%	DT 95%	GT	DT	DT 100%	DT 95%	GT	DT	DT 100%	DT 95%
Projection Profile	31	32	30	31	49	49	49	49	42	41	36	38	42	48	37	41
Run Length Smearing	31	32	30	31	49	57	0	0	42	46	37	40	42	46	35	41
Adaptive Run Length Smearing	31	31	31	31	49	54	0	0	42	43	39	39	42	41	38	41

IV. CONCLUSION

The algorithm based on the projection profiles cannot handle the images where the text lines are overlapping or touching. And it cannot divide those lines instead it will merge those lines. Run Length smearing algorithm fails if two text lines touch each other. Adaptive Run Length Smearing algorithm is not suitable for text lines with fully overlapped. We experimented these 3 algorithms on different types of documents like overlapping of text line MBRs, document contains different font sizes, document has non constant spaces between two text lines and good documents of English and Telugu scripts.

Projection profile shows 100% accuracy on well printed English documents and documents has non constant spaces. Run Length smearing algorithm performed 96% accuracy on overlapping documents, Adaptive Run Length smearing algorithm resulted 99% accuracy on English documents with Overlapping of components. These algorithms are not performed well on Telugu documents.

REFERENCES

[1] C. Bhagvati A. Negi and B. Krishna. An OCR system for telugu. Sixth International Conference

on Document Analysis and Recognition (IC-DAR'01

[2] Text Line Segmentation of Historical Documents:a Survey" Laurence Likforman-Sulem,Abderrazak Zahour, Bruno Taconet, International Journal on Document Analysis and Recognition, Springer, 2006.

[3] Script-Independent Text Line Segmentation in Freestyle Handwritten Documents" Yefeng Zheng2, David Doermann1, and Stefan Jaeger1

[4] Atul Negi, Chakravarthy Bhagvati and V.V Suresh Kumar, Non-linear normalization to improve telugu ocr.

[5] "Segmentation of historical machine-printed documents using Adaptive Run Length Smoothing and skeleton segmentation paths" N Nikolaou, M Makridis, B Gatos, N Stamatopoulos, N Papamarkos, Image and Vision Computing, Volume 28 Issue 24, April 2010.

[6] Atul Negi, Chakravarthy Bhagvati and V.V Suresh Kumar, Non-linear normalization to improve telugu ocr.

[7] S Mahesh Kumar Chakravarthy Bhagvati, Tanuku Ravi and Atul Negi. On developing high accuracy ocr systems for telugu and other Indian scripts. Language Engineering Conference, pages 1754–1759, 2002.

[8] N.Fakotakis,E.Kavallieratou and G.Kokkinakis. Skew angle estimation from printed and handwritten documents using the wigner-ville distribution. Image and Vision Computing, 20:813–824, 2002.

[9] Marti U., Bunke H. (2001), On the influence of vocabulary size and language models in unconstrained handwritten text recognition, Proc. of ICDAR'01, Seattle, pp. 260-265.

[10] Manmatha R., Srimal N (1999), Scale space technique for word segmentation in handwritten manuscripts,Proc. 2nd Int. Conf. on Scale Space Theories in Computer Vision, pp. 22-33.

[11] Shapiro V., Gluhchev G., Sgurev V. (1993), Handwritten document image segmentation and analysis, Pattern recognition Letters, 14:71-78.

[12] Zahour, A., Taconet, B., Mercy, P., Ramdane, S. (2001), Arabic hand-written text-line extraction, Proceedings of the 6th ICDAR, Seattle, pp. 281–285.

[13] Antonacopoulos A., Karatzas D (2004), Document Image analysis for World War II personal records, First Int. Workshop on Document Image Analysis for Libraries, DIAL'04, Palo Alto, pp. 336-341

◈◈◈