Greek Polytonic OCR based on Efficient Character Class Number Reduction

B. Gatos, G. Louloudis and N. Stamatopoulos

Computational Intelligence Laboratory,
Institute of Informatics and Telecommunications,
National Research Center "Demokritos",
153 10 Athens, Greece
{bgat, louloud, nstam}@iit.demokritos.gr

Abstract—**Recognition of document images having Greek polytonic (multi accent) characters is a challenging task due the large number of existing character classes (more than 270). In this paper, we propose a novel OCR framework for the recognition of machine-printed Greek polytonic documents that is based on combining five different recognition modules in order to have a small number of classes (around 30) in each module. One recognition module is used for accent recognition while four recognition modules are used for the recognition of characters belonging to different horizontal text zones. The proposed system also includes the following stages: a) pre-processing, b) text dewarping, text line and text baseline detection, c) accent and character detection and d) combination of accent and character recognition results. Extended experiments have been conducted in order to record the performance of the proposed OCR system, of all involved recognition modules as well as of the accent detection stage.**

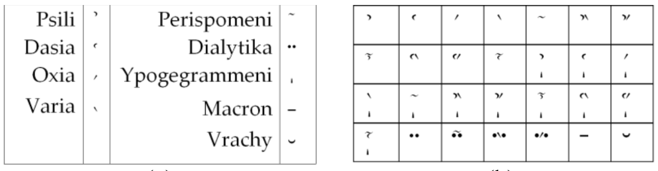
OCR, Greek polytonic characters; Class number reduction; Word baseline detection

I. Introduction

Although the accurate recognition of Latin machine-printed text is now considered largely a solved problem, recognition of scripts having a large number of characters is still the subject of active research. Chinese and Asian scripts are some examples that involve a large number of character classes and attract the attention of several researchers [1,2]. Greek polytonic (multi accent) scripts have a large variety of diacritic marks and, as a result, a large number of character classes (more than 270). Due to that, Greek polytonic documents cannot be successfully processed by current OCR technologies. Some approaches that use general purpose OCR engines for the recognition of Greek polytonic documents are mainly based on intense training [3] or post-processing [4] without significant successful results. Taking into consideration that the Greek polytonic system was used from around 200 BC to modern times until 1982, we observe that a large amount of scanned Greek documents still remains without full text search capabilities. To this end, we have focused our research efforts towards an OCR engine for Greek polytonic documents that reduces the number of involved character classes by combining different recognition modules each one with a small number of classes (around 30). One recognition module is used for accent recognition while four recognition modules are used for the recognition of characters belonging to different horizontal text zones. The proposed system includes five distinct stages: a) pre-processing (binarization and text column detection), b) text dewarping, text line and text baseline detection, c) accent and character detection, d) accent and character recognition, and e) combination of accent and character recognition results.

II. The Greek Polytonic System

The Greek polytonic system includes 9 diacritic marks (Fig.1a). Some of these marks are combined and as a result we have a total of 28 different diacritic mark combinations that may appear above or below Greek characters (Fig. 1b). The total number of Greek characters is 49 (25 lower case and 24 upper case). These characters are combined with the diacritic marks and as a result we have a total of 272 character classes (including numbers and special symbols) (see Fig. 2).


Psili
Dasia
Oxia
Varia
Perispomeni
Dialytika
Ypogegrammeni
Macron
Vrachy

(a) (b)

Figure 1. a) Diacritic marks and b) their combinations.

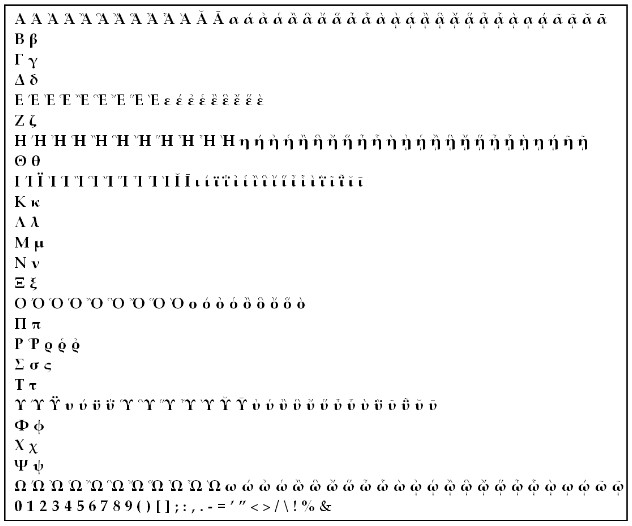
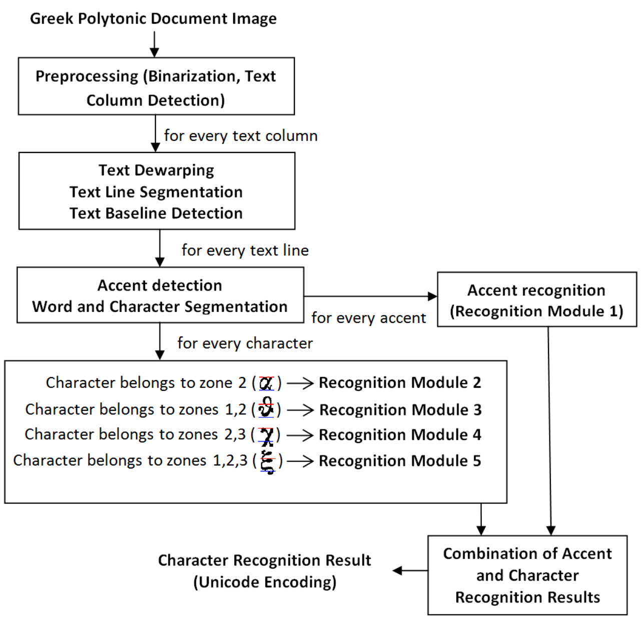

Β β
Γ γ
Δ δ
Ζ ζ
Θ θ
Κ κ
Λ λ
Μ μ
Ν ν
Ξ ξ
Π π
Σ σ ς
Τ τ
Φ φ
Χ χ
Ψ ψ
0 1 2 3 4 5 6 7 8 9 ( ) [ ] ; : , . - = ' " < > / \ ! % &

Figure 2. Greek polytonic characters.

III. The Proposed OCR Framework

The flowchart of the proposed OCR framework is shown in Fig. 3 and detailed in this Section. The input to our system is a gray scale, color or b/w, single or multi-column document image containing Greek polytonic text. The output is a Unicode encoded text file.



Greek Polytonic Document Image
Preprocessing (Binarization, Text Column Detection)
for every text column
Text Dewarping
Text Line Segmentation
Text Baseline Detection
for every text line
Accent detection
Word and Character Segmentation
for every accent
Accent recognition (Recognition Module 1)
for every character
Character belongs to zone 2 (α) → Recognition Module 2
Character belongs to zones 1,2 (ἀ) → Recognition Module 3
Character belongs to zones 2,3 (χ) → Recognition Module 4
Character belongs to zones 1,2,3 (ξ) → Recognition Module 5
Combination of Accent and Character Recognition Results
Character Recognition Result (Unicode Encoding)

Figure 3. Flowchart of the proposed OCR framework.

A. Preprocessing

Let I_x and I_y be the width and the height of the document image, respectively. At this stage, we first binarize the input image using the adaptive binarization technique of [5] and then detect the document text columns based on a background run-length processing technique. We first tag all background (white) pixels that belong to long vertical white runs (with length $>I_y/3$). Then, we un-tag all white pixels that belong to very short horizontal white runs (with length $<I_x/100$). At a next step, we trace all components formed by the tagged pixels in the vertical zone between $I_x/4$ and $3*I_x/4$. If there exist a component of significant height ($>I_y/2$) then this denotes a column separation region. Additionally, if the column separation region touches long horizontal white runs on the top or the bottom of the image then horizontal cuts are also performed. Image tagging for column detection is demonstrated in Fig. 4a. In this example, the image after tagging is divided into 3 sub images.

B. Text Dewarping - Text Line Segmentation – Text Baseline Detection

For every detected text column we first proceed to text dewarping in order to facilitate the application of further processing stages. We employ the two-step dewarping approach of [6]. According to this approach, a coarse dewarping is first accomplished with the help of a transformation model which maps the projection of a curved surface to a 2D rectangular area. At a second step, fine dewarping is achieved based on word segmentation information. An example of the text column dewarping result is demonstrated in Fig. 4b.

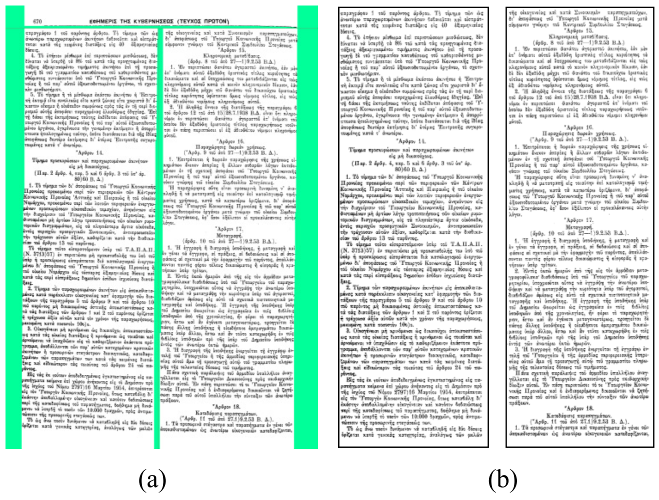


Figure 4. (a) Image tagging for text column detection and (b) text column detection result after applying dewarping.

At a next step for every dewarped text column we proceed to text line segmentation following the Hough Transform based approach of [7]. Then, for every text line we detect the text baselines as follows:

Let $L(x,y)$ be a text line image array having 1s for foreground and 0s for background pixels, L_x and L_y be the width and the height of the text line image, respectively. Let L_h the average character height of the image that equals to the average height of all connected components of the original image. Also, let $f(x_1,x_2,y_1,y_2)$ the number of foreground pixels in the text line image block (x_1,y_1) -(x_2,y_2). The text baseline can be approximated in the limits of y_b and $y_b + L_h$ where y_b corresponds to the y-value that maximizes function $f(1, L_x,y,y+ L_h)$. For a better approximation, we divide every text line image into 4 vertical zones and calculate the upper baseline limit B_u as follows:

$$B_u(x) = \begin{cases} \underset{y\in[1..L_y-L_h]}{argmax}\, f\left(0,\frac{L_x}{4},y,y+L_h\right) & \text{if } x \in [1 \ldots \frac{L_x}{4}] \\ \underset{y\in[1..L_y-L_h]}{argmax}\, f\left(\frac{L_x}{4},\frac{L_x}{2},y,y+L_h\right) & \text{if } x \in [\frac{L_x}{4} \ldots \frac{L_x}{2}] \\ \underset{y\in[1..L_y-L_h]}{argmax}\, f\left(\frac{L_x}{2},\frac{3L_x}{4},y,y+L_h\right) & \text{if } x \in [\frac{L_x}{2} \ldots \frac{3L_x}{4}] \\ \underset{y\in[1..L_y-L_h]}{argmax}\, f\left(\frac{3L_x}{4},L_x,y,y+L_h\right) & \text{if } x \in [\frac{3L_x}{4} \ldots L_x] \end{cases} \quad (1)$$

The lower baseline limit B_l is defined as B_l (x)= B_u (x)+ L_h. An example of text baseline detection is given in Fig.5.

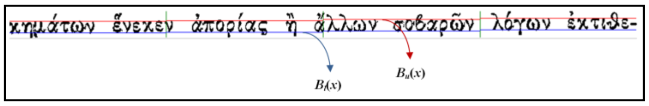


Figure 5. An example of text baseline detection where upper and lower baseline limits $B_u(x)$ and $B_l(x)$ are shown.

C. Accent detection – Word and Character Segmentation

At this stage, we first examine all connected components of every text line image L in order to detect accents. We aim to detect all diacritic mark combinations shown in Fig.1 except the ones having the diacritic mark “ypogegrammeni” which lie below the characters. At a first step, we join together all pairs of connected components that (a) overlap vertically and their horizontal distance is less than $L_h/4$ or (b) overlap horizontally and their vertical distance is less than $L_h/4$. In this way we connect neighboring components that belong to diacritic mark combinations or to broken diacritic marks. If the bounding box of a connected component is defined by co-ordinates (cx_1,cy_1) - (cx_2,cy_2), then this component is considered as accent if the following two conditions are both satisfied:

$$C1 = (cx_2 - cx_1 > L_h/10) \text{ AND } (cy_2 - cy_1 > L_h/10) \qquad (2)$$

$$C2 = (cy_2 < B_u(cx_1) \text{ OR } ((cy_2 < B_u(cx_1) + L_h/4) \text{ AND } (cx_2 - cx_1 < cy_2 - cy_1)) \qquad (3)$$

Condition $C1$ denotes that the connected component is large enough not to be considered as noise. Condition $C2$ denotes that the connected component belongs to the upper character zone (is upper than the already detected baseline) or partially belongs to the text baseline while it is vertically elongated. An accent detection example is shown in Fig.6.

At a next step we remove all detected accents from the text line image L and proceed to character segmentation based on the methodology described in [8]. According to this methodology, we calculate the skeleton of foreground and background pixels and detect all possible segmentation paths by linking the feature points on the skeleton of the image and its background. Skeleton feature points correspond to fork, end and corner points. Then, we use a set of rules in order to select the best character segmentation paths. A character segmentation example is given in Fig.7. Words are defined if the distance between consecutive characters is more than $0.7*L_h$.

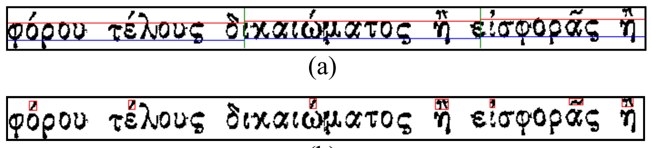


Figure 6. Part of a text line where (a) the text baseline and (b) the detected accents are shown.

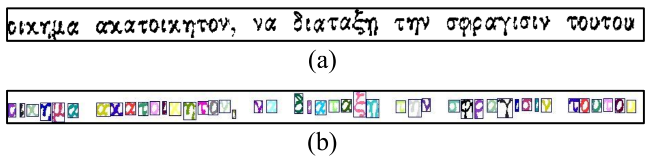


Figure 7. A character segmentation example. (a) Text line image after accent removal. (b) Character segmentation result.

D. Recognition Modules

Every detected character is first examined if it belongs to the three horizontal text zones that are defined by the text baseline. Zone 1 is the area upper to the baseline, zone 2 is the area that belongs to the baseline and zone 3 is the area below the baseline. A character is considered that belongs to a certain zone if at least 15% of its pixels are located within this zone. In this way we form four characters categories: a) characters that belong only to zone 2, b) characters that belong to zones 1 and 2, c) characters that belong to zones 2 and 3, and d) characters that belong to zones 1, 2 and 3. Fig.8 shows some examples of characters that belong to those categories.

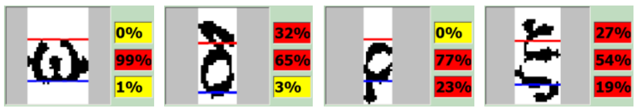


Figure 8. Example of characters that belong to different categories together with the pixel percentage for each zone.

In order to recognize the characters of the four above mentioned categories as well as the accents already detected in the previous stage, we build five recognition modules: Recognition Module 1 for accent recognition and Recognition Modules 2 – 4 for the recognition of characters in each category. Based on the geometry of the patterns, for every classifier we use different size normalization: 30x30 for Recognition Module 1, 30x40 for Recognition Module 2, 30x60 for Recognition Modules 3 and 4, and 30x80 for Recognition Module 5. The features we use are based on zoning (see also [9]) and are calculated by dividing the patterns to 10x10 windows. The total number of features is 9 for Recognition Module 1, 12 for Recognition Module 2, 18 for Recognition Modules 3 and 4, and 24 for Recognition Module 5. For all Recognition Modules we employ a K-NN classifier (K=3).

E. Final Recognition Result

The final character recognition result is extracted by combining accent recognition (Recognition Module 1) and character recognition (Recognition Modules 2, 3, 4 or 5) results. An example of this procedure is shown in Fig. 9. We use Unicode encoding in order to save the text produced.

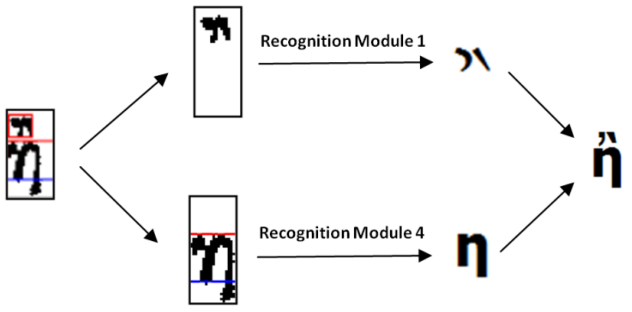


Figure 9. An example of the final character recognition result.

IV. Experimental Results

The experiments addressed in this work are based on a randomly selected set of images obtained from the Hellenic National Printing House which is responsible for publishing digital copies of the Laws and Presidential decrees of the Greek State [10]. The publication year of these documents ranges between 1950 and 1965. A document image sample is shown in Fig. 4.

Three different experiments were conducted. The first experiment aims to measure the ability of the accent detection methodology to correctly identify accents. The second experiment aims to test the accuracy of all involved recognition modules while the third experiment measures character and word accuracy of the final text produced by the proposed OCR framework.

A. Accent Detection Evaluation

In order to measure the accuracy of the proposed accent detection methodology we manually annotated all accents appearing on 10 document images of our set. An accent was considered as detected only if there was a significant overlap with the result produced by the accent detection methodology. The overlap is expressed by the intersection over union metric $IOU = \frac{A \cap B}{A \cup B}$ where A and B denote the bounding box areas of a manually annotated and detected accent, respectively [11]. The IOU metric ranges from 0 to 1, where 1 corresponds to exact matching. A threshold T (set to 0.95) is used in order to decide whether the accent in the ground truth and the detected entity match sufficiently. The efficiency of the proposed accent detection methodology is measured in terms of Precision, Recall and F-measure metrics. If N is the number of ground truth accents, M is the number of the detected accents and $o2o$ is the number of exact matches we calculate Precision, Recall and F-measure metrics as follows:

$$Precision = \frac{o2o}{M}, \; Recall = \frac{o2o}{N}$$
$$Fmeasure = \frac{2 * Presicion * Recall}{Precision + Recall} \tag{4}$$

As it is presented in Table I, the F-measure for the accent detection procedure is more than 97%.

Most of the errors encountered by the methodology are mainly due to noise or broken characters (see Fig. 10).

TABLE I. Accent Detection evaluation results

N	*M*	*o2o*	*Recall*	*Precision*	*Fmeasure*
5530	5670	5437	98.32%	95.89%	97.09%

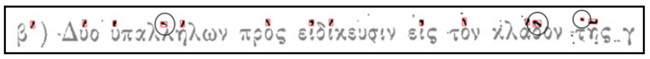

β') Δύο ὑπαλλήλων πρὸς εἰδίκευσιν εἰς τὸν κλάδον τῆς γ

Figure 10. Example of wrongly identified accents (in cirles).

B. Recognition Modules Evaluation

In order to evaluate the performance of all involved recognition modules we used three representative document images from our set. We extracted the accents as well as the characters that belong to several horizontal zones following the segmentation procedure described in the previous section. Then, we manually labeled all correctly segmented accents and characters with the correct class identifier. Following this procedure, we manually labeled 1572 accents and 8418 characters.

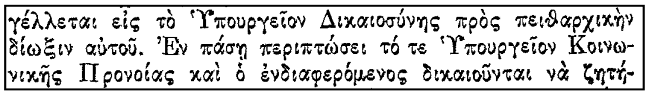
We used a 3-fold cross-validation process according to which the data from one document is used for testing and the remaining data for the other two documents are used for training. Table II illustrates the average recognition rates for all recognition modules. It can be observed that all recognition modules perform well achieving a recognition accuracy that ranges from 95% to 100%.

TABLE II. Evaluation of Recognition Modules

Recognition Module	**Number of classes**	**Number of characters**	**Recognition Rates**
1	12	1572	95.36%
2	17	6092	98.26%
3	30	620	95.32%
4	11	1665	99.64%
5	5	41	100.0%

C. OCR Accuracy Evaluation

The final experiment concerns the accuracy of the proposed OCR framework. We compared the text produced from the OCR framework with the correct text which was manually created for 7 representative document images of our set (see Fig. 11).


γέλλεται εἰς τὸ Ὑπουργεῖον Δικαιοσύνης πρὸς πειθαρχικὴν
δίωξιν αὐτοῦ. Ἐν πάσῃ περιπτώσει τό τε Ὑπουργεῖον Κοινω-
νικῆς Προνοίας καὶ ὁ ἐνδιαφερόμενος δικαιοῦνται νὰ ζητή-

(a)

γέλλεται εἰς τὸ Υπουργεῖον Δικαιοσύνης πρὸς πειθαρχικὴν
δίωξιν αὐτοῦ- Εν πάσῃ περιπτώσει τό τε Υπουργεῖον Κοινω-
νικῆς Προνοίας καὶ ὁ ἐνδιαφερόμενος δικαιοῦνται νὰ ζητή.

(b)

γέλλεται εἰς τὸ Ὑπουργεῖον Δικαιοσύνης πρὸς πειθαρχικὴν
δίωξιν αὐτοῦ. Ἐν πάσῃ περιπτώσει τό τε Ὑπουργεῖον Κοινω-
νικῆς Προνοίας καὶ ὁ ἐνδιαφερόμενος δικαιοῦνται νὰ ζητή-

(c)

Figure 11. (a) Part of the original document, (b) OCR result and (c) correct text that was manually created.

We used as training set the character and accent database created in the previous experiment. We measured the accuracy of the OCR framework in terms of Character Accuracy and Word Accuracy [12] that are calculated as follows:

$$Character\ Accuracy = \frac{\#characters - \#cerrors}{\#characters}$$
$$Word\ Accuracy = \frac{\#words - \#werrors}{\#words} \qquad (5)$$

where *#characters/#words* correspond to the number of characters/words in the ground truth text and *#cerrors /#werrors* corresponds to the number of character/word errors. A word is considered as an error if it has at least one character error. As it is presented in Table III, the proposed OCR framework achieves a character recognition rate of 90% and a word recognition rate of 63%. We observe that although the involved classifiers achieve higher character recognition accuracy results (see previous experiment) we have a recognition rate of 90% due to errors introduced mainly during the accent and character detection stages.

TABLE III. CHARACTER AND WORD ACCURACY

Entity	***#entities***	***#errors***	***Accuracy (%)***
Character	22896	2269	90.09%
Word	3245	1211	62.68%

In Table IV we present the recognition accuracy for some representative character classes. It is observed that using the proposed accent - character separation methodology we can achieve high recognition results even for character classes that look very similar, e.g. small letters with different accents.

TABLE IV. REPRESENTATIVE CHARACTER CLASS ANALYSIS

Class	***Total***	***Missed***	***% Right***
α	1175	46	96.09%
ά	240	17	92.92 %
ὰ	172	12	93.02 %
ἀ	174	22	87.36%
ε	748	48	93.58%
ἐ	219	29	86.76 %
έ	215	24	88.84 %
ο	1690	122	92.78%
ὸ	187	18	90.37 %
ό	150	36	76.00 %
ω	426	30	92.96 %
ῶ	180	20	88.89 %
ώ	72	9	87.50%

V. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a novel OCR framework for the recognition of Greek polytonic documents that is based on combining several recognition modules in order to have a small number of classes in each module. It includes a pre-processing stage for binarization and text column detection, text dewarping, text line and text baseline detection, accent and character detection, accent and character recognition, and, finally, combination of accent and character recognition results. Extended experiments have been conducted in order to record the performance of the proposed OCR system, of all involved recognition modules as well as of the accent detection stage. We have observed that a) accent detection procedure is more than 97% accurate, b) all recognition modules perform well achieving a recognition accuracy that ranges from 95% to 100% and c) the final OCR framework achieves a character recognition rate of 90% and a word recognition rate of 63%. The 10% of characters that are not correctly recognized are missed mainly due to errors introduced during the accent and character detection stages.

Our aim was to demonstrate the effectiveness of an integrated framework that we propose for the recognition of Greek Polytonic document images. To this end, we evaluated the performance of the integrated system as well as of the main involved components. At a next step, we will focus to improve the performance of all components giving a priority to the accent and character detection stages.

ACKNOWLEDGMENT

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

REFERENCES

[1] N. Wang, “Printed Chinese Character Recognition Based on Pixel Distribution Probability of Character Image”, IEEE Trans. Intelligent Information Hiding and Multimedia Signal Processing, Vol. 00, pp. 1403-1407, 2008.

[2] U. Pal and B. B. Chaudhuri, “Indian script character recognition: a survey”, Pattern Recognition, Vol. 37, Issue 9, pp. 1887-1899, 2004.

[3] http://groups.google.com/group/ocropus/web/ocropus-for-polytonic-greek

[4] F. Boschetti, M. Romanello, A. Babeu, D. Bamman, and G. Crane, “Improving OCR Accuracy for Classical Critical Editions”, Proc. 13th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2009), pp. 156-167, 2009.

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

[6] N. Stamatopoulos, B. Gatos, I. Pratikakis, S.J. Perantonis, “Goal-oriented Rectification of Camera-Based Document Images”, IEEE Trans. on Image Processing, DOI: 10.1109/TIP.2010.2080280, 2010.

[7] G. Louloudis, B. Gatos, I. Pratikakis, C. Halatsis, "Text line and word segmentation of handwritten documents", Pattern Recognition, Vol. 42, Issue 12, pp. 3169-3183, 2009.

[8] N. Nikolaou, M. Makridis, B. Gatos, N. Stamatopoulos, N. Papamarkos, "Segmentation of historical machine-printed documents using Adaptive Run Length Smoothing and skeleton segmentation paths", Image and Vision Computing, Vol. 28 , Issue 4, pp. 590-604, 2010.

[9] T. Konidaris, B. Gatos, K. Ntzios, I. Pratikakis, S. Theodoridis and S. J. Perantonis, "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.

[10] http://www.et.gr/

[11] C. Wolf, J. Jolion, “Object count/area graphs for the evaluation of object detection and segmentation algorithms”, International Journal on Document Analysis and Recognition, Volume 8, Number 4, pp. 280–296, 2006.

[12] S. Rice, Measuring the Accuracy of Page-Reading Systems, PhD thesis, University of Nevada, Las Vegas, 1996.