Proceedings of the Fourth IASTED International Conference
Signal Processing, Pattern Recognition, and Applications
February 14–16, 2007, Innsbruck, Austria
ISBN Hardcopy: 978-0-88986-646-1 / CD: 978-0-88986-647-8

HYBRID OFF-LINE OCR FOR ISOLATED HANDWRITTEN GREEK CHARACTERS

G. Vamvakas[1,2], B. Gatos[1], I. Pratikakis[1], N. Stamatopoulos[1,2], A. Roniotis[2], S.J. Perantonis[1]

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

[2]Department of Informatics
and Telecommunications
University of Athens, Greece

ABSTRACT
In this paper, we present an off-line OCR methodology for isolated handwritten Greek characters mainly based on a robust hybrid feature extraction scheme. First, image pre-processing is performed in order to normalize the character images as well as to correct character slant. At the next step, two types of features are combined in a hybrid fashion. The first one divides the character image into a set of zones and calculates the density of the character pixels in each zone. 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. For the classification step Support Vectors Machines (SVM) are used. The performance of the proposed methodology is demonstrated after testing with the CIL database (handwritten Greek character database), which was created from 100 different writers.

KEY WORDS
Handwritten OCR, Hybrid Features, Handwritten Character Database

1. Introduction

Optical character recognition (OCR) is one of the most successful applications of automatic pattern recognition. The OCR systems attempt to facilitate the every-day use of computers in the transformation of large amount of documents, either printed or handwritten, into electronic form for further processing. Nowadays, the recognition of printed isolated characters is performed with high accuracy. However, the recognition of handwritten characters still remains an open search problem. In general, character recognition procedure consists of two steps: (a) feature extraction where each character is represented as a feature vector and (b) classification of the vectors into a number of classes [1]. To develop high in at these two points. To get the outer profiles, for each y value, the outermost x values of each contour half is selected. To get the inner profiles, for each y value, the innermost x values are selected. Horizontal profiles can accuracy handwritten character recognition systems, researchers have taken great efforts searching for efficient feature representation and classifiers with good generalization ability.

Selection of a feature extraction method is probable the single most important factor in achieving high recognition performance [2]. Due to the nature of handwriting with its high degree of variability and imprecision, obtaining these features is a difficult task. A feature extraction algorithm must be robust enough that for a variety of instances of the same symbol, similar feature sets are generated, thereby making the subsequent classification task less difficult [3]. In the literature, feature extraction methods have been based on two types of features: statistical and structural [4]. Representation of a character image by statistical distribution takes care of style variations to some extent. This method is used for reducing the dimension of the feature set providing high speed and low complexity [5]. On the other hand, characters can be represented by structural features with high tolerance to distortions and style variations.

The most common statistical features used for character representation are: (a) zoning, where the character is divided into several zones and features are extracted from the densities in each zone, (b) projections, where the two-dimensional image (2-D) is represented as one-dimensional signal (1-D) [6] and (c) crossings and distances [5]. For example, contour direction features measure the direction of the contour of the character [7], which are generated by dividing the image array into rectangular and diagonal zones and computing histograms of chain codes in these zones. In [2], a feature extraction method which is based on horizontal and vertical profiles of the contour of the characters is presented. First the horizontal projection is computed locating the uppermost and lowermost pixels of the contour. The contour is split be extracted in a similar fashion, starting by dividing the contour in upper and lower halves. These features, although independent to noise and deformation, depend on rotation. Another example is presented in [8], where two sets of features are extracted using crossings and distances. The first one consists of the number of transitions from background to foreground pixels along vertical and horizontal lines through the character image and the second one calculates the distances of the first image pixel detected from the upper and lower boundaries, of the image, along vertical lines and from the left and right boundaries along horizontal lines.

Structural features are based on topological and geometrical properties of the character, such as maxima and minima, cusps above and below a threshold, cross points, branch points, strokes and their directions, inflection between two points, horizontal curves at top or bottom, etc. One of the most popular techniques for structural feature extraction is coding which is obtained by mapping the strokes of a character into o 2-D parameter space, which is made up the codes. Another method for structural feature representation is representation by graphs. The character is first partitioned into a set of topological primitives, such as strokes, loops, cross points etc. and, these primitives are represented using attributed or relational graphs [5].There are two kinds of image representation by graphs. The first kind uses the coordinates of the character shape [9]. The second one is an abstract representation with nodes corresponding to the strokes and edges corresponding to the relationships between the strokes [10].

For the recognition of handwritten Greek characters the only existing approach is based on structural features [11]. A 280-dimension vector is extracted consisting of histograms and profiles. The well known horizontal and vertical histograms are used in combination with the radial histogram, out-in radial and in-out radial profiles. Assume that each character is normalized to a 32x32 matrix. Consider that the value of the element in the *m*-th row and the *n*-th column of the character matrix is given by a function *f*:

$$f(m,n) = a_{mn} \qquad (1)$$

where a_{mn} takes binary values (i.e., 0 for white pixels and 1 for black pixels). The horizontal histogram H_h of the character matrix is the sum of black pixels in each row (i.e., 32 features):

$$H_h(m) = \sum_n f(m,n) \qquad (2)$$

Similarly, the vertical histogram H_v of the character matrix is the sum of black pixels in each column (i.e., 32 features):

$$H_v(n) = \sum_m f(m,n) \qquad (3)$$

The radial histogram is defined as the sum of foreground pixels on a radius that starts from the center of the image and ends up at the border. The radial histogram values are calculated with a step of 5 degrees (i.e., 72 features):

$$H_r(\phi) = \sum_{i=1}^{16} f(\|16 - i\sin\phi\|, \|16 + i\cos\phi\|) \qquad (4)$$
$$\phi = 5*k, \ \ \mathrm{k} \in [0,72)$$

The value of the out-in radial profile is defined as the position of the first foreground pixel found on the radius that starts from the periphery and goes to the center of the character image forming an angle φ with the horizontal axis. The out-in radial profile values are calculated with a step of 5 degrees (i.e., 72 features):

$$P_{oi}(\phi) = \left\{ \begin{array}{l} I : \sum_{i=16}^{I-1} f(\|16 - i\sin\phi\|, \|16 + i\cos\phi\|) \equiv 0 \\ \& f(\|16 - \mathrm{I}\sin\phi\|, \|16 + \mathrm{I}\cos\phi\|) \equiv 1 \end{array} \right\} \qquad (5)$$
$$\phi = 5*k, \ \mathrm{k} \in [0,72)$$

Similarly, the value of the in-out profile is defined as the position of the first foreground pixel found on the radius that starts from center of the character image and goes to the periphery forming an angle φ with the vertical axis. The in-out radial profile values are calculated with a step of 5 degrees (i.e., 72 features):

$$P_{io}(\phi) = \left\{ \begin{array}{l} J : \sum_{i=16}^{J-1} f(\|16 - i\sin\phi\|, \|16 + i\cos\phi\|) \equiv 0 \\ \& f(\|16 - J\sin\phi\|, \|16 + J\cos\phi\|) \equiv 1 \end{array} \right\} \qquad (6)$$
$$\phi = 5*k, \ \mathrm{k} \in [0,72)$$

Thus, a 280-dimension vector is extracted for each character, as presented in Fig. 1:

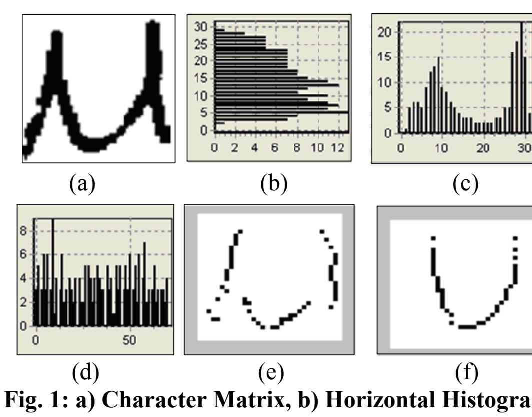


Fig. 1: a) Character Matrix, b) Horizontal Histogram, c) Vertical Histogram d) Radial Histogram, e) Radial out-in Profile, f) Radial in-out Profile.

In this paper, we present a hybrid statistical feature extraction method for isolated handwritten Greek characters, which is based on a combination of features extracted from dividing the character image into zones and the projections of the upper, lower, left and right profiles. The experimental results indicate the effectiveness of our approach. We mainly compare our method with the only existing method for recognizing handwritten Greek characters, which is described in [11]. The remaining of the paper is organized as follows. In Section 2 our approach is presented while experimental results are discussed in Section 3. Finally, conclusions are drawn in Section 4.

2. Proposed Method

2.1. Pre-processing

Before the feature extraction algorithm takes place, we first normalize all binary character images to a NxN matrix. After normalization, slope correction is performed, which is based on [12]. The dominant slope of the character is found from the slope corrected character which gives the minimum entropy of a vertical projection histogram. The vertical histogram projection is calculated for a range of slope correction angles, α_i, where the angle is given in $\pm\,\theta$. A slope correction range of θ=60^o appears to cover all writing styles. The slope of the character α_m, is found from:

$$\alpha_m = \min_{a\in\pm\theta} H \tag{7}$$

$$H = -\sum_{i=1}^{N} P_i \log P_i \tag{8}$$

where N is the number of the columns in the vertical projection histogram and P_i is the probability of a foreground pixel appearing in column i. The character is then corrected by α_m using (see Fig. 2):

$$x' = x - y\tan(a_m) \tag{9}$$

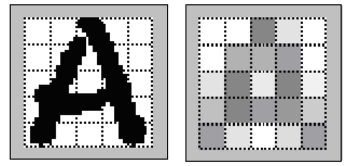
$$y' = y \tag{10}$$

2.2. Feature Extraction

In our approach we employ two types of features. The first one divides the character image into a set of zones and calculates the density of the character pixels in each zone. 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.

Let *im(x,y)* be the character image array having 1s for foreground and 0s for background pixels, x_{max} and y_{max} be the width and the height of the character image.

In the case of features based on zones, the image is divided into horizontal and vertical zones. For each zone,



(a) (b)

Fig. 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.

we calculate the density of the character pixels (see Fig. 2). Let Z_H and Z_V be the total number of zones formed in both horizontal and vertical direction. Then, features based on zones $f^{\,z}(i)$, i=0… $Z_H Z_V$ -1 are calculated as follows:

$$f^{\,z}(i) = \sum_{x=x_s(i)}^{x_e(i)} \sum_{y=y_s(i)}^{y_e(i)} im(x,y) \tag{11}$$

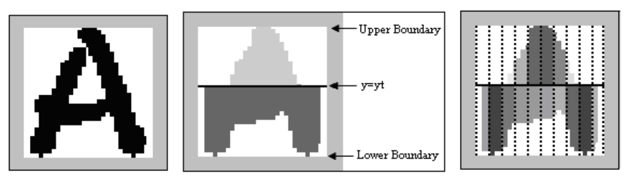
where,

$$x_s(i) = \left(i - \left\lfloor \frac{i}{Z_H} \right\rfloor Z_H\right)\frac{x_{\max}}{Z_H},\; x_e(i) = \left(i - \left\lfloor \frac{i}{Z_H} \right\rfloor Z_H + 1\right)\frac{x_{\max}}{Z_H}$$

$$y_s(i) = \left\lfloor \frac{i}{Z_H} \right\rfloor \frac{y_{\max}}{Z_V},\; y_e(i) = \left(\left\lfloor \frac{i}{Z_H} \right\rfloor + 1\right)\frac{y_{\max}}{Z_V}$$

In case of features based on character (upper/lower) profile projections, the character image is divided into two sections separated by the horizontal line $y = y_t$ (see Eq. 12):

$$y_t = \frac{\sum_x \sum_y im(x,y)\cdot y}{\sum_x \sum_y im(x,y)} \tag{12}$$



(a) (b) (c)

Fig. 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.

Upper/lower profiles (Eq. 13, 14) 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 Fig. 3).

$$y_{up}(x) = y_t - y_0, \tag{13}$$

$$\text{where } y_0 = \begin{cases} y_t, \text{ if } \sum_{y=0}^{y_t} im(\mathrm{x,y}) = 0 \\ y : (im(\mathrm{x,y}) = 1 \,\&\, \mathrm{y} = \min(\mathrm{y_i})), \mathrm{y_i} \in [0, \mathrm{y_t}], \text{else} \end{cases}$$

$$y_{lo}(x) = y_0 - y_t, \tag{14}$$

$$\text{where } y_0 = \begin{cases} y_t, \text{ if } \sum_{y=y_t}^{y_{\max}} im(\mathrm{x,y}) = 0 \\ y : (im(\mathrm{x,y}) = 1 \,\&\, \mathrm{y} = \max(\mathrm{y})), \mathrm{y_i} \in [\mathrm{y_t}, \mathrm{y_{max}}], \text{else} \end{cases}$$

Let P_V be the total number of blocks formed in each produced zone (upper, lower). For each block, we calculate the area of the upper/lower character profiles denoted as in the following:

$$f^{P}{}_{up\_ar}(i) = \sum_{x=x_s(i)}^{x_e(i)} y_{up}(x) \tag{15}$$

$$f^{P}{}_{lo\_ar}(i) = \sum_{x=x_s(i)}^{x_e(i)} y_{lo}(x) \tag{16}$$

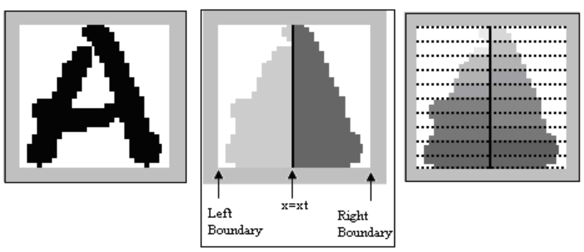
where,

$$x_s(i) = (i - \left\lfloor \frac{i}{P_V} \right\rfloor P_V)\frac{x_{\max}}{P_V}\ ,\ x_e(i) = (i - \left\lfloor \frac{i}{P_V} \right\rfloor P_V + 1)\frac{x_{\max}}{P_V}$$

and $i=0 \,..\, P_V-1$. Fig. 3 illustrates the features extracted from a character image using projections of character profiles.

In case of features based on character (left/right) profile projections, the character image is divided into two sections separated by the vertical line $x = x_t$ (see Eq. 17)

$$x_t = \frac{\sum_x \sum_y im(x,y)\cdot x}{\sum_x \sum_y im(x,y)} \tag{17}$$



(a) (b) (c)

Fig. 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.

Left/right character profiles (Eq. 18,19) are computed by considering, for each image column, the distance between the vertical line $x=x_t$ and the closest character pixel to the left/right boundary of the character image (see Fig. 4):

$$x_{le}(y) = x_t - x_0, \tag{18}$$

$$\text{where } x_0 = \begin{cases} x_t, \text{ if } \sum_{x=0}^{x_t} im(x,y) = 0 \\ x : (im(x,y) = 1 \,\&\, x = \min(x_i)),\ x_i \in [0, x_t], \text{else} \end{cases}$$

$$x_{ri}(y) = x_0 - x_t, \tag{19}$$

$$\text{where } x_0 = \begin{cases} x_t, \text{ if } \sum_{x=x_t}^{x_{\max}} im(x,y) = 0 \\ x : (im(x,y) = 1 \,\&\, x = \max(x_i)),\ x_i \in [x_t, x_{\max}], \text{else} \end{cases}$$

Let R_V be the total number of blocks formed in each produced zone (left, right). For each block, we calculate the area of left/right character profiles denoted as in the following:

$$f^{R}{}_{le\_ar}(i) = \sum_{y=y_s(i)}^{y_e(i)} x_{le}(y) \tag{20}$$

$$f^{R}{}_{ri\_ar}(i) = \sum_{y=y_s(i)}^{y_e(i)} x_{ri}(y) \tag{21}$$

where,

$$y_s(i) = (i - \left\lfloor \frac{i}{R_V} \right\rfloor R_V)\frac{y_{\max}}{R_V}\ ,\ y_e(i) = (i - \left\lfloor \frac{i}{R_V} \right\rfloor R_V + 1)\frac{y_{\max}}{R_V}$$

and $i=0 \,..\, R_V-1$. Fig. 4 illustrates the features extracted from a character image using projections of character profiles.

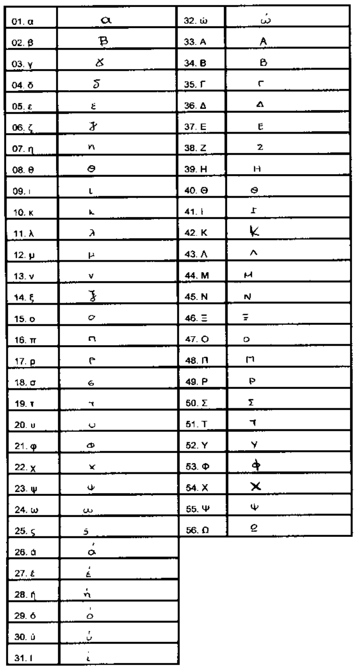
The overall calculation of the proposed feature vector is given in Eq. 22. The corresponding feature vector length equals to $Z_H Z_V + 2P_V + 2R_V$.

$$f(i) = \begin{cases} f^{Z}(i) = \sum_{x=x_s(i)}^{x_e(i)} \sum_{y=y_s(i)}^{y_e(i)} im(x,y),\ i = 0 \,..\, Z_H Z_V - 1 \\ f^{P}{}_{up\_ar}(i) = \sum_{x=x_s(i-Z_H Z_V)}^{x_e(i-Z_H Z_V)} y_{up}(x),\ i = Z_H Z_V \,..\, Z_H Z_V + P_V - 1 \\ f^{P}{}_{lo\_ar}(i) = \sum_{x=x_s(i-Z_H Z_V+P_V)}^{x_e(i-Z_H Z_V+P_V)} y_{lo}(x),\ i = Z_H Z_V + P_V + \ldots + Z_H Z_V + 2P_V - 1 \\ f^{R}{}_{le\_ar}(i) = \sum_{y=y_s(i-Z_H Z_V+2P_V)}^{y_e(i-Z_H Z_V+2P_V)} x_{le}(y),\ i = Z_H Z_V + 2P_V + \ldots + Z_H Z_V + 2P_V + R_V - 1 \\ f^{R}{}_{ri\_ar}(i) = \sum_{y=y_s(i-Z_H Z_V+2P_V+R_V)}^{y_e(i-Z_H Z_V+2P_V+R_V)} x_{ri}(y),\ i = Z_H Z_V + 2P_V + R_V + \ldots + Z_H Z_V + 2P_V + 2R_V - 1 \end{cases} \tag{22}$$

3. Experimental Results

For our experiments, we have used the CIL Greek characters database. This database was created by handling a number of forms (see Fig. 5 below) that include 56 Greek characters. Initially, 100 writers are requested to fill these forms and afterwards, a scanner converts them to digital binary images. The last stage deals with character extraction: Firstly, it detects line and column intersections of each form and thus, approximates the coordinates of cells. However, very slight inclination of forms, due to scanning, may cause seriously erroneous cell detection. Then, in order to avoid this, each cell's co-ordinates are re-defined, by scanning cell upwards and downwards until border lines are found. Lastly, characters are detected within cells and are sorted within the CIL database.

Every writer filled 5 forms, resulting in a data base of 28,000 isolated and labeled characters. Each class represents a character and consists of 500 variations of this character (56x500=28,000). After the size normalization step some characters such as the upper-case 'O' and the lower-case 'o', are considered to be the same. So, for having meaningful results we merged these two classes into one, by randomly selecting 500 characters from both classes. This was done to a total of 10 pair of classes. Table 1 shows which classes are merged. This concluded in having 46 classes with 500 patterns in each class and the database now has 46x500=23,000 characters



01. α	α	32. ώ	ώ
02. β	β	33. Α	Α
03. γ	γ	34. Β	Β
04. δ	δ	35. Γ	Γ
05. ε	ε	36. Δ	Δ
06. ζ	ζ	37. Ε	Ε
07. η	η	38. Ζ	Ζ
08. θ	θ	39. Η	Η
09. ι	ι	40. Θ	Θ
10. κ	κ	41. Ι	Ι
11. λ	λ	42. Κ	Κ
12. μ	μ	43. Λ	Λ
13. ν	ν	44. Μ	Μ
14. ξ	ξ	45. Ν	Ν
15. ο	ο	46. Ξ	Ξ
16. π	π	47. Ο	Ο
17. ρ	ρ	48. Π	Π
18. σ	σ	49. Ρ	Ρ
19. τ	τ	50. Σ	Σ
20. υ	υ	51. Τ	Τ
21. φ	φ	52. Υ	Υ
22. χ	χ	53. Φ	Φ
23. ψ	ψ	54. Χ	Χ
24. ω	ω	55. Ψ	Ψ
25. ς	ς	56. Ω	Ω
26. ά	ά		
27. έ	έ		
28. ή	ή		
29. ό	ό		
30. ύ	ύ		
31. ί	ί		

Fig. 5: Sample of the forms used for the CIL Greek Characters Data Base.

Moreover, 1/5 of each class was used for testing and the 4/5 for training. So the used database was split into a training set of 18,400 characters (400x46=18,400) and a testing set of 4,600 characters (100x46= 4,600).

Table 1: Merged Classes

	Upper-case	Lower-case
1	Ε	ε
2	Θ	θ
3	Κ	κ
4	Ο	ο
5	Π	π
6	Ρ	ρ
7	Τ	τ
8	Φ	φ
9	Χ	χ
10	Ψ	ψ

As it has already been described in Sections 2 we have used a size normalization step followed by a slope correction step before feature extraction. During the normalization step, the size of the normalized character images used is x_{max}=60 and y_{max}=60. In the case of features based on zones, the character image is divided into five (Z_H =5) horizontal and five (Z_V=5) vertical zones forming a total of twenty-five (25) blocks with size *12x12* (see Fig. 2). Therefore, the total number of features is 25. In the case of features based on character (upper/lower) profile projections we keep the same size of the normalized image, while the image is divided into ten (10) vertical zones (P_V =10) (see Fig. 3). Consequently, the total number of features equals to twenty (20). Similarly, the normalized image divided into ten (10) horizontal zones (R_V=10) (see Fig. 4). Therefore, the total number of features equals to twenty (20). Combination of features based on zones and features based on character profile projections led to the feature extraction model (Eq. 22) that uses a total of sixty-five (65) features.

In the particular binary classification problem classification step was performed using two well-known classification algorithms, Euclidean Minimum Distance Classifier (EMDC) [13] and Support Vector Machines (SVM) [14].

Formally, the support vector machines (SVM) require the solution of an optimization problem, given a training set of instance-label pairs (x_i, y_i), i=1,…, m, where $x_i \in R^n$ and $y_i \in \{1,-1\}^m$. The optimization problem is defined as follows:

$$\begin{aligned} &\min_{\omega,b,\xi} \quad \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{m}\xi_i \\ &subject\ \ to\ \ y_i(\omega^T\phi(x_i)+b) \geq 1-\xi_i \\ &\qquad \xi_i \geq 0 \end{aligned} \qquad (23)$$

Regarding the classification step, SVM was used in conjunction with the Radial Basis Function (RBF) kernel, a popular, general-purpose yet powerful kernel, denoted as:

$$K(x_i, x_j) \equiv \exp(-\gamma \|x_i - x_j\|^2) \qquad (24)$$

Furthermore, a grid search was performed in order to find the optimal values for both the variance parameter (γ) of the RBF kernel and the cost parameter (C) of SVM (see Eq. 23).

Table 2 depicts the recognition rates (%) for the proposed hybrid method, the single feature methods and the method described in [11] which is the only method reported in the literature for the recognition of Greek handwritten characters. These recognition rates achieved after combining slope correction with either single features of both feature extraction schemes. We can draw several conclusions. First, as it can been easily seen the best performance, for both classifiers, is achieved in the case of using an additive fusion resulted after the combination of slope correction preceding the hybrid feature extraction scheme. Moreover, our approach seems to have better results although the total number of features used (65) is smaller than the one in [11] (280). Finally, the SVM achieved considerably higher recognition rates than the EMDC.

4. Conclusion

This paper proposes an off-line OCR system for isolated handwritten Greek characters that is based on an additive fusion resulted after a novel combination of two different modes of character image normalization (size normalization and slope correction) and robust hybrid feature extraction.

Table 2: Experimental Results

Pre-processing	Features			Number of features	Classifier		Recognition Rate (%)
Slope Correction	Method in [11]	Hybrid			EMDC	SVM	
		Zones	Projections				
	√			280	√		80.65%
√	√			280	√		81.56%
		√		25	√		82.67%
			√	40	√		77.23%
		√	√	65	√		83.21%
√		√		25	√		84.30%
√			√	40	√		79.06%
√		√	√	65	√		**85.08%**
	√			280		√	87.52%
√	√			280		√	88.62%
		√		25		√	86.76%
			√	40		√	87.60%
		√	√	65		√	92.34%
√		√		25		√	87.74%
√			√	40		√	88.04%
√		√	√	65		√	**92.91%**

We compared the proposed approach with the method described in [11], which is the only work in recognizing Greek handwritten characters. Our approach achieved higher rates of recognition.

Our future research will focus on exploiting new features as well as fusion methods to further improve the current performance.

Acknowledgements

This research is carried out within the framework of the Greek Ministry of Research funded R&D project POLYTIMO [15] which aims to process and provide access to the content of valuable historical books and handwritten manuscripts.

References

[1] S. Britto, R. Sabourin, F. Bortolozzi, C. Y. Suen, Foreground and Backround Information in an HMM-Based Method for Recognition of Isolated Characters and Numeral Strings, *9th International Workshop on Frontiers in Handwriting Recognition (IWFHR-9)*, October 26-29, 2004, Kokubunji, Tokyo, Japan, pp 371-376.

[2] O. D. Trier, A. K. Jain, T.Taxt, Features Extraction Methods for Character Recognition – A Survey, *Pattern Recognition , 29*(4) : 641-662, 1996.

[3] J. A. Fitzgerald, F. Geiselbrechtinger, and T. Kechadi, Application of Fuzzy Logic to Online Recognition of Handwritten Symbols, *9th International Workshop on Frontiers in Handwriting Recognition (IWFHR 9)*, Tokyo, Japan, pp. 395-400, October 26-29, 2004.

[4] V.K. Covindan, A.P. Shivaprasad, Character Recognition – A Review, *Pattern Recognition, 23*(7), 1990, pp. 671- 683.

[5] N. Arica and F. Yarman-Vural, An Overview of Character Recognition Focused on Off-line Handwriting, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 31*(2), 2001, pp. 216 - 233.

[6] Y. Tao, Y. Y. Tang, The feature extraction of Chinese character based on contour information, *in Proc. 5thInt. Conf. Document Anal. Reognit.* , Bangalore, India, 1999, pp. 637 – 640.

[7] K. M. Mohiuddin and J. Mao, A Comprehensive Study of Different Classifiers for Handprinted Character Recognition. Pattern *Recognition, Practice IV*, pp. 437-448, 1994.

[8] J. H. Kim, K. K. Kim, C. Y. Suen, Hybrid Schemes Of Homogeneous and Heterogeneous Classifiers for Cursive Word Recognition, *7th International Workshop on Frontiers in Handwriting Recognition,* September 11-13 2000, Amsterdam, ISBN 90-76942-01-3, Nijmegen: International Unipen Foundation, pp 433 - 442s.

[9] H. Cheng, W. H. Hsu, M. C. Kuo, Recognition of handprinted Chinese characters via stroke relaxation, *Pattern Recognition, 26*(4), pp 579 – 593, 1993.

[10] W. Lu, Y. Ren, C. Y. Suen, Hierarchical attributed graph representation and recognition of handwritten Chinese characters, *Pattern Recognition, 24*(7) pp. 617 – 632, 1991.

[11] E. Kavallieratou, N. Fotakis, G Kokkinakis, Handwritten Character Recognition Based on Structural Characteristics, icpr, p. 30139, *16th International Conference on Pattern Recognition (ICPR'02) - Volume 3*, 2002.

[12] R. Buse, Z.Q. Liu, and T. Caelli, A Structural and Relational Approach to Handwritten Word Recognition, *IEEE Trans. Systems, Man, and Cybernetics, Part B, 27*(5), pp. 847-861, Oct. 1997.

[13] Theodoridis, S., and Koutroumbas, K., *Pattern Recognition*,(Academic Press, 1997).

[14] Cortes C., and Vapnik, V., Support-vector network, *Machine Learning*, vol. 20, pp. 273-297, 1997.

[15] POLYTIMO project, http://iit.demokritos.gr/cil/Polytimo, 2006.