Ancient document recognition using fuzzy methods

J. M. C. Sousa, J. R. Caldas Pinto, C. S. Ribeiro and J. M. Gil
Technical University of Lisbon, Instituto Superior Técnico
Dept. of Mechanical Engineering, GCAR/IDMEC
1049-001 Lisbon, Portugal

Abstract— **This paper proposes an optical character recognition system based on fuzzy logic for ancient printed documents. The recognition process consists of two stages: training with collected character image examples and classification of new character images. The proposed OCR builds fuzzy membership functions from oriented features extracted using Gabor filter banks. Results on a significant test led to a character recognition success rate of 88%.**

Keywords: Fuzzy OCR, character recognition, old documents.

I. Introduction

Optical character recognition (OCR) is a practical application of state-of-the-art image processing and pattern recognition developments [9]. Uses of OCR include digital document archiving, printed text search and automated form processing. Current communication facilities could allow broad and public distribution of vast libraries of books, newspapers, magazines and all kinds of printed media, if quality, cost-effective OCR procedures are available for mass digitizing.

While modern printed text can be recognized very accurately with commercially available software, performing OCR on more exotic material (such as gothic fonts, ancient typesets and handwriting) is currently and noticeably less successful [8].

This paper proposes a fuzzy recognizer specifically tailored to ancient documents and corresponding typesets. The proposed algorithm is a development of a handwriting word recognition system using fuzzy logic [1]. The use of fuzzy classification [13] improves results by providing larger tolerance for unstable typesetting and printing technologies.

The recognizer is based on an analytic perspective, i.e., it considers each character separately. Building a holistic recognizer able to handle nearly any text in full would require training with virtually every single word in a language, demanding enormous memory resources and taking an unacceptably long time to classify each word. Holistic recognition is far better suited for mass indexing by a few known, relevant words; generic OCR using such a system is unrealistic with current technology. Ancient documents are not so difficult as handwritten recognition, but clearly more difficult than standard font OCR. OCR is certainly a very useful tool to manipulate information of old documents in a digital format. However, OCR of ancient documents should take into account their specificities.

The recognition algorithm proposed in this paper is especially suited for old documents, and it works in two steps. The first step, training, considers sets of character images, known as character groups, and combines their dominant graphical features, resorting to Gabor filter banks to execute oriented feature extraction [2]. These composite images are then used to build fuzzy membership functions that, in a sense, describe the visual attributes of every character group [1]. The second step, classification, is where the actual recognition takes place. A new character image is processed by Gabor filters and normalized, and then it is compared to the training results. The closest match, dictated by a fuzzy decision maker, is returned as the most likely classification.

This paper is divided as follows. First, the global recognition system overview is presented in Section II. Then, the fuzzy recognizer is described in Section III. In this section both the training and the classification are described. The results for the proposed recognizer are presented in Section IV, where it is shown that the algorithms proposed in this paper are superior to one of the best commercial OCR packages. Finally, some conclusions are drawn in Section V.

II. System Overview

This section intends to give an introduction to the general functioning of the developed application. Figure 1 displays a diagram representing schematically the organization of the developed recognition system and the streamlined design connecting its components. The segmentation to obtain each individual character image is performed using the OCR package ABBYY FineReader Engine [12]. This package also provides methods to segment entire words, according to the computed geometric information, as well as its own OCR output, which was used as a base of comparison with the fuzzy recognition system. Note that this is one of the most advanced commercial packages for OCR. Moreover, we are using a development license containing the most recent advances in the software. Therefore, our implementation is compared to the most recent state-of-the-art OCR techniques. The information obtained from the FineReader OCR is used to build a manually classified character database, which is applied in the training stage to build models for every known character. When a new image is given for recognition, the most likely classification is given to each segmented character, thus performing the intended OCR on the characters.

The proposed OCR recognizes characters instead of words, which was proposed in [1]. This change reduces resource requirements; character images are smaller, so processing takes less time and the size of the training data structure is easily

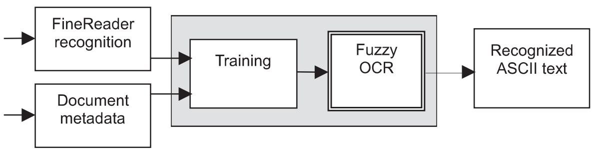


Fig. 1. Diagram of the fuzzy document recognizer.

handled by modern hardware. Further, recognizer parameters and thresholds are configured differently as well, adjusted to provide a finer and more comprehensive analysis of character features. Other original changes include the disabling of the alignment process and an additional aspect ratio classification factor. They are described and highlighted in Section III.

III. Fuzzy Recognizer

The recognition process requires a previous training step, followed by the intended character classification. These two stages are presented in this section.

A. Training

The training process is performed in two steps: oriented feature extraction, where the dominant features of a character are extracted using Gabor filter banks, and membership function generation, which are generated for each character group and for each orientation, based on the extracted features of the training images.

The dominant features of a character consist of what is more common not to change between typing styles, as the long vertical stroke in the b's and t's, for example. In this paper their extraction is performed through Gabor filter banks, which allow oriented feature extraction. Each filter, oriented at a given angle ϕ, extract the features of a character.

1) Feature extraction: The oriented feature extraction is performed using Gabor filters. The Gabor filter is a typical wavelet that offers localized operations [6]. The result of the filtering can be used to extract local information from regions of the image, in time and frequency domains, and it can achieve minimum uncertainty in both of them [2]. The Gabor filter are defined in a spacial (x, y) and in a frequency (u, v) domains as, respectively:

$$g(x,y) = \exp\left\{-\pi\left(\frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2}\right)\right\} \times \exp\left\{2\pi j\left(u_0 x + v_0 y\right)\right\} \tag{1}$$

$$G(u,v) = \exp\left\{-\pi\left[(u' - u_0')^2\sigma_x^2 + (v' - v_0')^2\sigma_y^2\right]\right\} \tag{2}$$

where

$$\begin{aligned} x' &= x\cos\phi + y\sin\phi & y' &= -x\sin\phi + y\cos\phi \\ u' &= u\cos\phi + v\sin\phi & v' &= -u\sin\phi + v\cos\phi \\ u_0' &= u_0\cos\phi + v_0\sin\phi & v_0' &= -u_0\sin\phi + v_0\cos\phi \\ u_0 &= f\cos\theta & v_0 &= f\sin\theta \\ j &= \sqrt{-1} & f &= \sqrt{u_0^2 + v_0^2}. \end{aligned}$$

Eq. (2) represents a 2-D Gaussian centered at (u_0, v_0) in the frequency domain. The parameters σ_x and σ_y are the standard deviations of the 2-D Gaussian, determining the frequency and orientation bandwidths of the filter. The angle ϕ defines the Gabor wavelet direction, and the angle $\theta = \phi + 90^\circ$ defines the wavelet orientation. The frequency f determines the distance to the origin of the image frequency spectrum.

The spacing between the Gabor angles $\Delta\phi$ is an important parameter. However, its value is not critical in optimizing the shape of the Gabor filters for extracting parts. These parameters are character dependent, which means that the estimation has to take into account, e.g., the size of the characters and the thickness of the writing [2]. For this reason, usually, their values are selected on a trial-and-error basis. For this paper, 12 wavelet directions were considered, from 0° to 180°, as in [1].

Features are extracted by first applying a discrete Fourier transform to the image. The resulting output is processed to avoid spurious features. Because this output is complex, the power image is used; the parts oriented at the direction ϕ_i have a larger intensity than the parts oriented away from that direction. The image is normalized to produce consistent results among distinct cases, by resizing each image to the largest dimension among each character group.

The word recognition in [1] required a time-consuming alignment algorithm in order to match extracted word features among word group samples. This procedure is needed, especially for handwritten text, to compensate for variations in character spacing and shape, and normalize word bounds. This procedure is not implemented in this paper. Notice that characters instead of words are being recognized. Character alignment can produce heavy distortion when feature match is not effective. A single character has a smaller number of dominant features; printing flaws, common in this context, complicate feature matching even further. Character segmentation is also tight and accurate from the beginning. This simplification reduces the computational effort significantly.

As each training sample of the same character contains essentially the same extracted features structure, the major structural components can be established by adding the standardized images together, which form the composite image for each trained character. Character images are classified manually; ancient characters are labelled as their present-day equivalent, therefore solving the problem of generating standard ASCII text from these occurrences. An example of original images is given in Fig. 2. Figure 3 presents the image

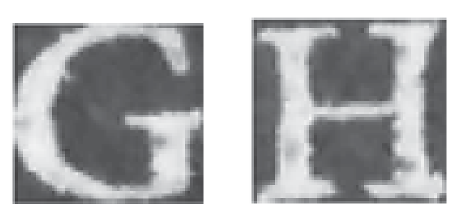


Fig. 2. Original character images.

resulting from applying Gabor filters with 90° orientation.

Besides the information in the Gabor filters, in this paper

Fig. 3. Feature extraction using Gabor filters with 90° orientation.

information regarding image aspect ratios for the characters is also stored. The aspect ratio of a given image I is defined as

$$ar(I) = \frac{w(I)}{h(I)} \tag{3}$$

where $w(I)$ is the image width and $h(I)$ is the image height. The average aspect ratio ar_j for each character group j is defined as:

$$ar_j = \frac{\sum_{k=1}^{N_j} ar(I_{jk})}{N_j} \tag{4}$$

where I_{jk} is the kth sample image for character group j and N_j is the total number of images for group j. Aspect ratio values ar_j are used in classification as further assistance in the identification of unknown characters.

2) Membership function generation: Before the classification process can take place, a set of fuzzy membership functions is generated for each character group and for each orientation, based on the extracted features of the training images. The membership functions intend to provide a description of the image features for use within the recognition algorithm.

Each training sample of a character group C_j contains the main features structure. Thus, a composite image R_{ij}, can be constructed for each point (pixel) (x, y) as follows:

$$R_{ij}(x, y) = \frac{1}{N_j} \sum_{k=1}^{N_j} C_{ijk}(x, y), \tag{5}$$

where i indicates an orientation (as e.g. the ones of the Gabor filters), k is a sample image of the jth character group.

The membership functions are represented by twisted trapezoids. This shape is defined by two oriented bounded rectangles. This shape provides a better fit to the shape of the data than 2–D trapezoids [1]. Upper and lower thresholds must be defined for the upper and lower rectangles, based on the composite images R_{ij}. The upper, H_u, and the lower, H_l, thresholds are used in the binarization of the image [11], thus finding the upper and lower boundaries for each feature. These thresholds are determined by:

$$H_u = c_u \max\{R_{ij}(x, y)\} \tag{6}$$

$$H_l = c_l \max\{R_{ij}(x, y)\}, \tag{7}$$

where $R_{ij}(x, y)$ was defined in (5), and constants c_u and c_l are set at 0.4 and 0.25, respectively. These values were determined empirically in [1] and are quite successful, but they can also be based on a standard binarization method [10] for more demanding cases and greater robustness. The two bounding rectangles are determined around the extracted features, as proposed in [14], corresponding to each intensity threshold. The bounding rectangles resulting from the upper thresholds defined in (6) for the features extracted using Gabor filters, presented in Fig. 3, are shown in Fig. 4.

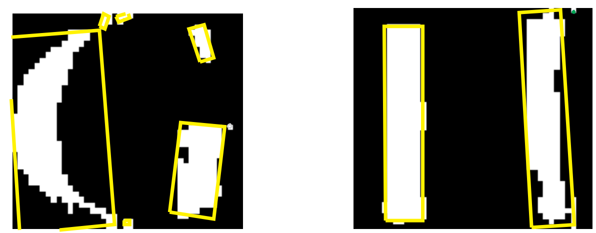


Fig. 4. Bounding rectangles around the extracted features.

Each rectangle pair is used to build a partial membership function. Its value $\mu(x, y)$ is 1 in the inner rectangle area and zero outside the outer rectangle. The vertices of a rectangle pair are linked based on minimization of the Euclidean distance. The intermediate function values are interpolated, forming a twisted trapezoidal shape [5], [1]. To do so, the

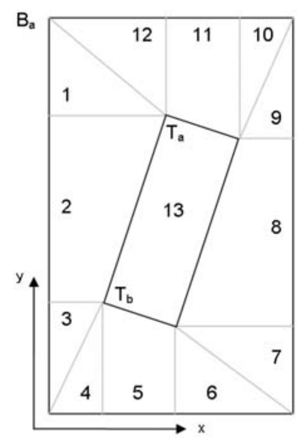


Fig. 5. Domain regions of the generated membership function.

function domain is divided into 13 regions, as represented in Fig. 5. Region 1 in this figure has the following membership value:

$$\mu(x, y) = \frac{x - x_{B_a}}{x_{T_a} - x_{B_a}}, \tag{8}$$

where x and y are the coordinates of a given point, B_a, T_a and T_b are the points shown in Fig. 5, which are described by their respective coordinates x and y. Similar expressions can be found for regions 3, 4, 6, 7, 9, 10 and 12. The membership function for region 2 in Fig. 5 has the following values:

$$\mu(x, y) = \frac{(x - x_{B_a})(y_{T_b} - y_{T_a})}{(y - y_{T_a})(x_{T_b} - x_{T_a}) + (x_{T_a} - x_{B_a})(y_{T_b} - y_{T_a})}. \tag{9}$$

Similar interpolation rules are applied to regions 5, 8 and 11. Thus, the expressions for regions 3 through 12 are analogous and can be obtained by considering the rectangle vertices in sequence and swapping the coordinate pairs. In region 13, the membership value is naturally $\mu(x,y) = 1$.

The global membership function for a given orientation i, denoted as $a_{ij}(x,y)$, is defined as the maximum of the partial membership functions at each point:

$$a_{ij}(x,y) = \max\left(\mu_{ij\ell}(x,y)\right) , \qquad (10)$$

where j denotes a character group and ℓ is a membership function. Thus, in cases in where features overlap, (10) takes only into account the most relevant feature. These maximum values are found continuously during the membership function generation process, in order to minimize resource usage. The membership function generated for the letter 'G' and the orientation of the Gabor filter equal to 90° in Fig. 2 is presented in Fig. 6.

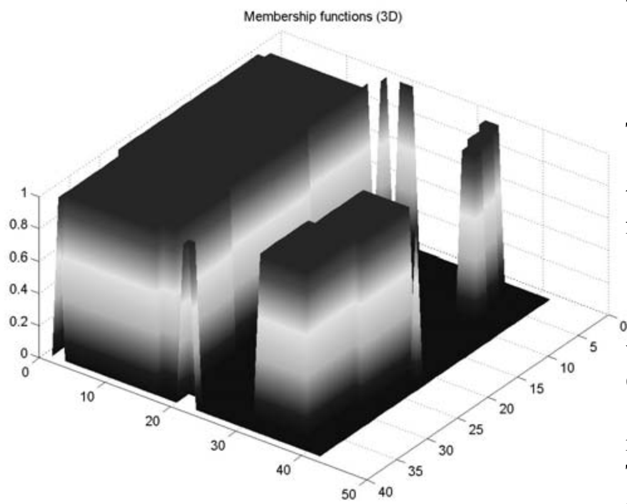


Fig. 6. Membership function generated from example in Fig. 2 for the character 'G' and 90° orientation of the Gabor filter.

B. Classification

Once the training phase is complete, it is possible to classify new unseen characters. The input for this stage is a character image C and the global membership functions built during the training phase described in Section III-A. First, the image is pre-processed in order to extract its features. Namely, it is filtered through Gabor filter banks and normalized, exactly as applied in the training phase. The purpose of this step is to match the features of the processed image to those of the training images, resulting in a set of feature images C_i. This set is compared, using a fuzzy decision process, to the training character groups and their respective membership functions.

A similarity rating can be computed between the test character and the membership functions of the training characters. A larger similarity should translate a bigger match between the character image being evaluated C and the training character images C_j. The largest similarity indicates the closest match, and the input character is classified as belonging to the character group with higher similarity value. This similarity S_{ij} is defined as:

$$S_{ij}(C) = S(C, \phi_i, C_j) \qquad (11)$$

i.e. S_{ij} is a function of the input character C to be classified, the angles of the Gabor filters ϕ_i and the training character groups C_j. The details of this calculation are defined in the following.

The similarity measure is calculated using a weighted average of the global membership functions $a_{ij}(x,y)$ defined in (10), and the intensity value of an input character C_i for orientation i, normalized to one, which is denoted as C'_i. Considering that the weight of a pixel (x,y) is denoted as $w_{ij}(x,y)$, the similarity is given by:

$$S_{ij} = \frac{\sum_{x,y} w_{ij}(x,y) a_{ij}(x,y) C'_i(x,y)}{\sum_{x,y} w_{ij}(x,y)} . \qquad (12)$$

The weights $w_{ij}(x,y)$ are assigned to each image point (x,y) for each orientation i and each character group j, to measure its influence, related mostly to the membership function values. The weights are calculated according to:

$$w_{ij}(x,y) = \begin{cases} C'_{ij}(x,y) & \text{if } a_{ij} = 0 \\ w'_{ij}(x,y) & \text{if } a_{ij} \neq 0 \end{cases} \qquad (13)$$

where $C'_{ij}(x,y)$ is a point in the normalized test character (location of the point (x,y) in the test character C'_{ij}).

The upper term in (13) assures that points where $C'_{ij}(x,y)$ is not zero but the membership function is, will be penalized. This happens when a feature in $C'_{ij}(x,y)$ does not match any orientation i for character group j. In this case, the value increases the denominator of (12), while not affecting the numerator, and therefore lowers the computed similarity rating.

The lower term in (13) is the rate of significance of point (x,y), and is denoted by $w'_{ij}(x,y)$. Considering that N_c is the total number of character groups, the weighting function is given by:

$$w'_{ij}(x,y) = \begin{cases} 0, & \text{if } N_+ = 0 \\ \frac{N_c + N_+}{N_+(N_c - 1)} \sum_{j=1}^{N_c} a_{ij}(x,y), & \text{if } N_+ > 0 \end{cases} \qquad (14)$$

where N_+ is the number of character groups, for each orientation and each point, with a positive membership grade $a_{ij}(x,y)$. It attempts to formalize the intuitive concept that point (x,y) is a distinguishing factor among character groups, when only a restricted set of groups has membership values for that point.

The similarity ratings proposed in (12) are determined considering the need to penalize the value of points with high $w_{ij}(x,y)$ but low or zero $a_{ij}(x,y)$ or $C'_i(x,y)$. In these cases, the image point has non-zero intensity outside the membership function area or low or zero intensity within the membership function area. It should be noted that the similarity measure in (12) is similar to the one proposed in [1]. However, that paper used a somewhat different and more complex formal notation, in part to account for the formal distinction between the various partial membership functions for each word and orientation. In this paper, the partial functions, each corresponding to a particular feature, were previously combined, which leads to a simpler notation and the improvement of computational resource usage.

Several methods can be used to aggregate the similarity values S_{ij} [3], from which the simple additive weighted method [7] is one of the most utilized; it was used in the holistic recognizer presented in [1]. In this paper we propose a different aggregation method which leads to better classification results. The decision is based on the normalized values for the global membership functions for a given orientation i, and a character group j, which is defined as v_{ij}:

$$v_{ij} = \frac{\sum_{x,y} a_{ij}(x,y)}{\sum_{i,x,y} a_{ij}(x,y)} . \tag{15}$$

This value is the ratio between membership function volume for a given orientation i and the total function volume for every orientation, within a character group. The rationale is that orientations with a relatively larger membership function volume should have a greater influence in the decision-making.

The similarity values are normalized, and denoted S'_{ij}, with respect to the orientations i:

$$S'_{ij} = \frac{S_{ij}}{\max_{i}(S_{ij})} \tag{16}$$

This paper introduces a new *aspect factor* $r_j(C)$, which compares the aspect ratio of the character image C with the average aspect ratio ar_j of character group j. Recall that the aspect ratio was defined in (3). The aspect factor is defined as:

$$r_j(C) = \min\left(\frac{ar(C)}{ar_j}, \frac{ar_j}{ar(C)}\right) . \tag{17}$$

The value of $r_j(C)$ is always smaller than 1, and decreases as the correspondence between the two aspect ratios decreases. This means that the match of an image with a given class is greater when the aspect ratios are more similar, as intuitively expected. This new factor increases recognition success.

Considering the normalized membership functions defined in (15), the normalized similarity values in (16) and the aspect factor proposed in this paper and defined in (17), the test character is classified as belonging to the character group identified by the j^* index, which is defined as:

$$j^* = \arg\max_j \frac{\sum_{i=1}^{N_c} v_{ij} S'_{ij}}{\sum_{i=1}^{N_c} v_{ij}} r_j(C) \tag{18}$$

Summations for j^* are performed over all orientations and each single character group. The test character is finally classified as belonging to the character group identified by the j^* index.

C. Recognizer algorithm

The general execution flow of the character recognition algorithm proposed in this paper is summarized next:

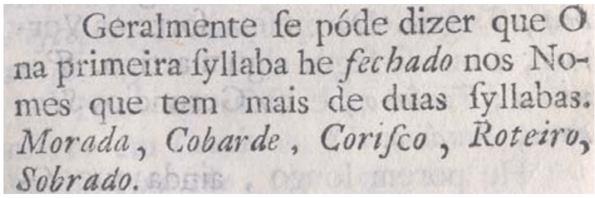
1) *For each character group, extract features:*
   a) *Apply Gabor filters to every sample, as defined in (1) and (2);*
   b) *Compute the average aspect ration for the character images using (4);*
   c) *Compose images per feature orientation;*
2) *For each character group, perform training:*
   a) *Perform two-level binarization using the thresholds in (6) and (7);*
   b) *Find feature-binding rectangles;*
   c) *Generate fuzzy membership functions using (8) and (9);*
3) *Perform classification of unknown images:*
   a) *Extract features as in training, see Step 1);*
   b) *Compute the weights to each pixel, as defined in (13);*
   c) *Compute the similarity matrix (12);*
   d) *Determine the aspect factor introduced in (17);*
   e) *Determine most likely classification of a new character in a character group using (18).*

IV. Results

A training character database was created from 1980 alphabetic characters classified manually. Recognition is possible once the training structure has been generated. In order to test various development options, the training data from this database is used in the classification of another previously identified 1580 character set, which works as a validation set.

Large representative test sets were sought so that the final results can be considered realistic and reliably convey the application performance in an actual common usage environment. Verifying the results achieved in the recognition tests requires a very time-consuming and thorough manual check of the output. Naturally, the tests actually executed were selected carefully within practical limits in order to be representative. The main test set consists of 20 pages acquired with variable scanning conditions, namely skewing and paper see-through, with both non-italic and italic text. It contains 1886 words consisting in 8034 characters, which is a significantly large test set, segmented by the FineReader engine. The source book [4] concerns Portuguese language orthography, providing a large variety of characters and formatting properties. Fig. 7 shows a sample paragraph.

Per-character results for this test set are summarized in Table I, were the fuzzy recognizer proposed in this paper is compared to the FineReader engine [12], one of the best commercial package for OCR. Both systems successfully classified most of the 8034 characters. The improvement introduced by the fuzzy recognizer is slight, although consistent. Many errors occur due to printing defects and strong similarity between certain key characters. Note however that the tests show that the fuzzy recognizer performance is superior to the FineReader



Geralmente ſe póde dizer que O na primeira ſyllaba he *fechado* nos Nomes que tem mais de duas ſyllabas. *Morada*, *Cobarde*, *Coriſco*, *Roteiro*, *Sobrado*.

Fig. 7. Sample paragraph from the 20 page test set.

engine, which is a remarkable feature, considering the relative simplicity of the proposed recognizer.

System	**20 page set success (%)**
FineReader engine	86.9
Fuzzy recognizer	88.0

TABLE I
PER-CHARACTER SUCCESS RATES.

An additional 12 page set from the same book discussing Portuguese language properties, is also used for some tests. It has diverse typesets and several printing problems, even though scanning quality is quite high. This test set has 1590 words.

The two test sets were used to evaluate the standard FineReader recognition and the fuzzy recognizer output. Table II shows the success rate for each of these cases in the two test sets. Recognition output was analyzed on a word basis; any word with at least one misclassified character is considered wrong; checking is case-insensitive and graphical accents are ignored.

System	20 page set (%)	12 page set (%)
FineReader engine	62.9	34.6
Fuzzy recognizer	64.1	35.6

TABLE II
PER-WORD SUCCESS RATES.

Per-word results can be considered unfair towards the FineReader and fuzzy recognizers, because these are character-based and not word-based. Most wrong words had few incorrect characters, explaining why per-character success rates are higher than per-word success rates. Note however that the fuzzy recognizer is again consistently better than the FineReader.

V. CONCLUSIONS

This paper proposed a character recognizer based on fuzzy pattern recognition. The system was designed to recognize old printed documents, containing several defects. Improvements of recognition results was noticeable with the fuzzy recognizer module. The fuzzy system achieved a success rate that is slightly better than a mature commercial software package.

Further work can include the development of an automatic parameter adjustment system based on measurable properties of the documents being processed, the definition of better word distance metrics, the introduction of more accurate heuristics and the development of an ancient word dictionary for spell checking.

ACKNOWLEDGEMENTS

This work was partly supported by: the "Programa de Financiamento Plurianual de Unidades de I&D (POCTI), do Quadro Comunitário de Apoio III"; the FCT project POSI/SRI/41201/2001; "Programa do FSE-UE, PRODEP III, Quadro Comunitário de Apoio III"; and program FEDER. We also wish to express our thanks to the Portuguese Bibioteca Nacional for their continuous support, which made possible this work.

REFERENCES

[1] R. Buse, Z. Q. Liu, and J. Bezdek. Word recognition using fuzzy logic. *IEEE Transactions on Fuzzy Systems*, 10(1):65–76, February 2001.

[2] R. Buse, Z. Q. Liu, and T. Caelli. A structural and relational approach to handwritten word recognition. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, 27(5):847–861, October 1997.

[3] S. J. Chen, C. L. Hwang, and F. P. Hwang. *Fuzzy Multiple Attribute Decision Making, Methods and Applications*. Springer-Verlag, Berlin, Germany, 1992.

[4] Álvaro Ferreira de Véra. *Orthographia ou modo para escrever certo na lingua Portuguesa*. Available at Biblioteca Nacional (National Library), Lisbon, Portugal, 17th century.

[5] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, editors. *Computer Graphics: Principles and Practice in C (2nd Edition)*. Addison-Wesley, Massachussets, USA, 1990.

[6] D. Gabor. Theory of communications. *J. Inst. Electr. Eng.*, 93:429–457, 1946.

[7] C. L. Hwang and K. Yoon. *Multiple Attribute Decision Making, Methods and Applications, A State-of-the-Art-Survey*. Springer-Verlag, Berlin, Germany, 1981.

[8] L. C. Jain and Beatrice Lazzerini, editors. *Knowledge-Based Intelligent Techniques in Character Recognition*. CRC Press, Boca Raton, Florida, 1999.

[9] Shunji Mori, Hirobumi Nishida, and Hiromitsu Yamada. *Optical Character Recognition*. Wiley Interscience, New York, 1999.

[10] N. Otsu. A threshold selection method from gray level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, January 1979.

[11] J. R. Parker, editor. *Algorithms for Image Processing and Computer Vision*. Wiley & Sons, New York, USA, 1998.

[12] ABBYY Software House. ABBYY FineReader. http://www.abbyy.com.

[13] J.M.C. Sousa and U. Kaymak. *Fuzzy Decision Making in Modeling and Control*. World Scientific Pub. Co., Singapore, 2002.

[14] G. Toussaint. Solving geometric problems with the rotating calipers. In *Proc. IEEE Mediterranean Electrotechnical Conference, MELECON'83*, pages A10.02/1–4, Athens, Greece, May 1983.