Unsupervised Font Clustering Using Stochastic Version of the EM Algorithm and Global Texture Analysis

Carlos Avilés-Cruz, Juan Villegas,
René Arechiga-Martínez, and Rafael Escarela-Perez

Universidad Autónoma Metropolitana - Azcapotzalco, Departamento de Electrónica,
Av. San Pablo 180 Col. Reynosa, C. P. 02200,
Distrito Federal, México
caviles@correo.azc.uam.mx

Abstract. An Unsupervised Font clustering technique is proposed in this work. The new approach is based on global texture analysis, using high order statistic features, Gaussian classifier and a stochastic version of the EM algorithm. The font recognition is performed by taking the document as a simple image, where one or several types of fonts are present. The identification is not performed letter by letter as with conventional approaches. In the proposed method a window analysis is employed to obtain the features of the document, using fourth and third order moments. The new technique does not involve a study of local typography; therefore, it is content independent. A detailed study was performed with 8 types of fonts commonly used in the Spanish language. Each type of font can have four styles that lead, to 32 font combinations. The font recognition with clean images is 100% accurate.

1 Introduction

Font recognition is a fundamental issue in the identification and analysis of documents. Khoubybari and Hull [2] took a document as an image, where clusters of word images were located within a reference word function base. The base font was chosen as the font that is more similar to the one being analyzed. In reference [3] a set of local detectors were employed to identify individual features of each font, such as height, width, thickness, base line, etc. Shi and Pavlidis [4] made recognition of fonts based on histogram properties of words, where inclination properties, histogram densities, etc. were measured. Zramdini and Ingold [5] show a statistical approach for the recognition of fonts based on their local topographic aspects. A similar approach was taken by Schreyer [6], where local attributes of textons were used (see [7] for the definition of textons and [8] as an alternate approach to textons). It can be seen that all these works are based on typographic aspects that were extracted with very local analysis instead of a global analysis. Only one author [9] was found to make use of global texture analysis. Gabor filters were tuned at different frequencies and orientations, leading to recognition results that are a function of pepper noise (degradation of recognition as a function of additive noise). Although the results given in [9] are good, there is room for substantial improvements. Experience shows that a global analysis may lead to good results in pattern recognition.

In this paper, the global analysis approach was followed. The use of high-order statistical moments (third and fourth order) and a principal-component analysis were proposed to characterize the textures of documents (fonts). As a method is unsupervised, the maximum of likelihood was used as a way to find the number of fonts. The likelihood was found via the expectation maximization algorithm (EM-algorithm). Once the mean value and variance/covariance matrix were estimated, those values are then passed to a standard statistical classifier, such as Bayes [10]. The purpose of this paper was to determine if the use of the method described here would result in better unsupervised font recognition. The approach followed is independent of document content and it is based on global texture analysis. The font identification process proposed in this work is summarized in the flow chart of fig. 1. The original image is pre-processed so as to create a uniform text block, which, in turn, is used to extract high-order statistical attributes. An analysis of principal components is then applied to eliminate linear redundancies, leading to a reduced number of features. Finally, the stochastic EM-algorithm get the number of different fonts and their parameters. For the identification task, the document (only test) is scanned window by window and estimated their high order features (third and fourth order moments); The features can now be categorized using a Bayes classifier. The structure of the paper is organized as follows. Section 2 gives details of the pre-processing stage, whereas the features used during the clustering process and the way to extract them are given in section 3. The definition and implementation of third and fourth order moments are included in section 4. The unsupervised classification method is discussed in section 5 and section 6 shows the efficacy of the proposed approach. Finally, future work and conclusions are presented in section 7.

Fig. 1. General scheme for font identification system

2 Pre-processing: Creation of a Uniform Block of Text

The text to be analyzed is contained within a JPG format file (the jpg format is converted to gray level image), which was used for the learning and identification stages. The text can include space characters or spaces between words, letters or lines. It is assumed that the document to be analyzed only contains text information, that is, no pictures or figures are in the document. In addition, characters may have different

sizes between words or lines. There can be two types of spaces within a line: a) spaces between characters, which are characteristic and unalterable of each type of font and b) spaces between words, which are irrelevant, as they do not provide any information about the type of font. The font information is extracted by going through the words in order to obtain a uniform block of text. The same procedure is applied to spaces between lines in order to eliminate them. The pre-processing stage is divided in four stages that are described in the following subsections.

2.1 Locating Text Lines

The location of a text line was made by calculating the horizontal projection profile (HPP) of the whole text, which was determined by adding over each line the intensity of the pixels that belong to it (gray levels). The values are then normalized with respect to the maximum value found. Figure (2b) shows an example of this procedure. The valleys between peaks correspond to blank spaces between text lines. The distance between two valleys gives the height of each text line. As a result, it is possible to locate and to determine the height of each text line (see fig. 2).

The printing area is 122 mm × 193 mm. The text should be justified to occupy the full line width, so that the right margin is not ragged, with words hyphenated as appropriate. Please fill pages so that the length of the text is no less than 180 mm.

Fig. 2. Example of pre-processing. (a) Original image, (b) HPP of the image, (c) image after line normalization, and (d) uniform text block after pre-processing

2.2 Text Line Normalization

A text can contain several types of fonts and different sizes for them. Therefore, it is necessary to normalize letters and words of different sizes to a standard one. Once a text line was located (see previous section), fonts were easily normalized to have them all to be of the same size. However, it is worth to mention that small font deformations could lead to small mistakes when the normalization stage was applied, for example an original letter “i” could appear as “l” or “1”.

2.3 Spacing Normalization

The normalization of vertical spacing was used to reduce the undesired influence of spaces on each line. In other words, it was used to eliminate spaces between words. The vertical projection profile (VPP) was calculated for each line, a valley between peaks corresponds to the width of each character or word. The normalization is used to obtain a predefined constant width. Figure 3 shows an example of this procedure.

Fig. 3. Example of spacing normalization. (a) Original text line, (b) VPP of line text, and (c) normalized line text

2.4 Text Padding

Since the text may not be justified, refilling of blank spaces was performed (at this stage) when the text did not ended with the rest of the lines. The option followed consisted on copying parts of the text of the preceding (or following) line, as there is a bigger probability that words be of the same type when they are close to each other. Figure 2d shows an example of a completely pre-processed text.

3 Feature Extraction

The uniform text obtained in the previous section, could be analyzed with any texture technique, evidently each technique is well adapted for texture type (i.e. fine grain

texture, coarse grain texture, etc). As it is remarked in the introduction section, only one paper using “global texture analysis”[9] was found and it is based on Gabor filters. In our project, we proposed to use high order statistics (third and fourth order moments) since in [8] Julesz is shown that when only first and second order moments are used it is not always possible to distinguish two textures. In the literature, there are techniques to analyze textures (in general texture not text [25],[26],[27] and [28]): Gabor filters [9],[11], co-occurrence matrices [12], [13],[29],[31], principal component analysis [14], [10], low-order statistical moments (mean value, standard deviation, correlation, etc.) [4], [14], [15], [24],[30], high-order statistical moments [16], [17], [15] and analysis with wavelet transforms [18], [19]. A method is given in reference [19] for the recognition of characters, but the recognition of the type of fonts is disregarded. In this project, statistical moments were used since it has been found that they have a very good performance with non-structured random textures [16], [17], [1], [15]. The goal of this paper is to report an other methodology to the optical font recognition problem based on high order statistics showing that our scheme improve the results obtained in [9]. The following subsections describe the theoretical and estimation aspects involved with third and fourth order moments.

Fourth and third order moments

Statistical moments represent an averaging process of the values (powered to order n) when a random variable is involved. Here, the original and pre-processed images were considered as two-dimensional arrays of a random variable of dimension $N \times N$. The random variable takes values from level 0 to 255, as the images were considered in gray levels quantized in 8 bits (gray levels are obtained from the jpg format). The images were obtained directly from a scanner or were digitally obtained from word processors or image converters.

Moments were calculated for the random variable X , which was identified with the image block. In addition, X is a vector obtained through matrix line linking. The definition of third-order central moments is given by:

$$m_3(i, j) = E[(X - \bar{X})_n * (X - \bar{X})_{n+i} * (X - \bar{X})_{n+j}] \quad (1)$$

where $E[\cdot]$ is the mathematical expectation or statistical average, $\bar{X}$ is the average value of the block to be processed, n is the central pixel whereas (i, j) represents the spaces between values of the random variable.

The estimator of (1) can be expressed as:

$$m_3(i, j) = \frac{1}{N} \sum_{n=1}^N ((x - \bar{x})_n * (x - \bar{x})_{n+i} * (x - \bar{x})_{n+j}) \quad (2)$$

where N is the total number of image pixels. It is worth mentioning that this estimator is biased. The values that (i, j) may get are $[0, 1, 2]$, such a small neighbor is chosen in order to reduce the huge number of possible combinations. When small neighborhoods are taken around the central pixel n . different combinations of eq. (1) can be developed by taking into consideration these small regions. For third order moments and taking into account all possible combinations, only six estimation are the most significant (between 9) when considering the non-redundant ones ($m_3(0,0)$, $m_3(0,1)$, $m_3(0,2)$, $m_3(1,1)$, $m_3(1,2)$ and $m_3(2,2)$).

Fourth-order central moments of a random variable X can be obtained from:

$$m_4(i, j, k) = E[(X - \bar{X})_n * (X - \bar{X})_{n+i} * (X - \bar{X})_{n+j} * (X - \bar{X})_{n+k}] \quad (3)$$

The average value of four pixels is taken in this case. The estimator of eq. (4) is given by:

$$m_4(i, j, k) = \frac{1}{N} \sum_{n=1}^N ((x - \bar{x})_n * (x - \bar{x})_{n+i} * (x - \bar{x})_{n+j} * (x - \bar{x})_{n+k}) \quad (4)$$

If small regions are considered around the central pixel n , the values that (i, j, k) may take in this case are $[0, 1, 2]$. As a result, ten non-redundant estimator expressions (the most significant ones) can be obtained¹: $m_4(0, 0, 0)$, $m_4(0, 0, 1)$, $m_4(0, 0, 2)$, $m_4(0, 1, 1)$, $m_4(0, 1, 2)$, $m_4(0, 2, 2)$, $m_4(1, 1, 1)$, $m_4(1, 1, 2)$, $m_4(1, 2, 2)$ and $m_4(2, 2, 2)$.

The estimations were performed with four orientations (0, 45, 90 and 135 degrees) in order to take into account the non-isotropic characteristics of the different font types (see fig. 4). Hence, 24 third-order moments result (4 orientations multiplied by 6 features per orientation). 40 moments were obtained for the fourth-order case (4 orientations multiplied by 10 features per orientation). The total number of estimated moments was 64 per window, which were obviously linearly redundant and excessive. A principal component analysis was applied to each moment in order to reduce attributes and linear relationships. The dimensions that retain the 99.99% of inertia were hold.

Fig. 4. Feature reduction by Principal Components Analysis (PCA) in order to withdraw linear redundancies. This scheme is also valid for fourth order moments

4 Font Clustering

The methodology applied in our processing image algorithm was based on to estimate features not over full image, instead of this, feature estimation was done over regions of images called “sub-image”. The estimated attribute arrays of each sub-image were the reference database to clustering font process (100 windows are taken randomly over full text). Fig. 5 depicts the principal component analysis of the high-order moments (third or fourth order) from their features, where the number of dimensions taken keeps 99.99% of inertia. This lead to a matrix composed of 800 vectors of dimension 24 for third-order moments, whereas 800 vectors of dimension 40 were

¹ Remarking that $m_4(0, 0, 1) = m_4(0, 1, 0) = m_4(1, 0, 0)$, $m_4(0, 0, 2) = m_4(0, 2, 0) = m_4(2, 0, 0)$ and so on.

obtained for fourth-order moments. 99.99% of inertia was achieved with three dimensions for both cases. As a result, a learning database, with 800 vectors of dimension 3 for each moment, results after post-processing. Unsupervised technique was used in order to determine the number of fonts and their parameters (mean value and variance/covarianza matrix). The stochastic EM algorithm was initialized to 32 functions (searching up to 32 different fonts types and styles), of course, if the fonts number is less than that, several functions will have the same parameters and as result only one is taken. Once parameters determined, a Bayes classifier was chosen to fulfill the classifications task.

Unsupervised Technique

The unsupervised technique is based in the likelihood theory, the foundation is to say, *a priori*, the data model is gaussian, so the total model is a mix of gaussian functions.

$$f(x|\Theta) = \sum_{k=1}^K P_k f_k(x|\theta_k) \quad (5)$$

Where

$$f_k(x|\theta_k) = \frac{1}{(2\pi)^{d/2} |\Sigma_k|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1} (x-\mu_k)} \quad (6)$$

P_k is the factor of mixing of model k ($0 < P_k < 1$) and $\sum_{k=1}^K P_k = 1$, θ_k is the vector of parameters $[\bar{\mu}_k, \bar{\Sigma}_k]$ and Θ is the vector of global parameters $[P_1, P_2, \dots, P_K; \theta_1, \theta_2, \dots, \theta_K]$ and d is the dimension of the sample $\bar{x}$.

For this model, two parameters will be calculated, the first one is the mean value and the second one is the variance. The “EM” (“Expectation-Maximization”) algorithm is a general technique for the estimation of those parameters. In this paper, the stochastic version of the EM algorithm was used [30],[31].

5 Results

Several tests were performed to validate the method proposed in the project. Eight different types of fonts were chosen: a) Arial, b) Bookman, c) Century Gothic, d) Courier, e) Comic Sans MS, f) Impact, g) Modern and h) Times New Roman, see figure 8. In addition, four different styles for each font were employed: a) Regular, b) Bold, c) Italic and d) Bold Italic, see figure 9. A total of 32 combinations of styles and fonts were considered. This particular set of fonts and styles was chosen in order to compare the results with those produced in reference [9]. It is interesting to mention that the fonts considered here were those given by the Microsoft Word Processor.

The images considered in this work were digitally generated by the Microsoft Word Processor and then converted to images. This way, it is possible to control all the fonts and their sizes accurately. A text was written using fonts of 12 point size at 300 dpi. Each image is 640×640 pixels in size for each type of font.

Finding the right size of each window is essential to make a good estimation, that is, the number of pixels that are necessary to reach convergence with the high-order

moments considered. This task was performed by making estimations of fourth and third order moments with different window sizes, starting from a 2x2 pixel size and ending with the maximum image size (640x640 pixel size in this work). If the whole image, the estimation was taken as the reference value, estimation errors can be calculated for smaller windows. Fig. 6 shows the estimation errors, where it can be seen that for a 32x32 window size, the error was less than 5%. Due to space constrains, Fig. 6 shows only up to 64x64 window size because, and however the error estimation remained on zero for the windows of size 64x64 to 640x640.

Once defined the size of the estimation (32x32 pixels, allowing an error less than 5% on convergence), the methodology described on Fig. 1 and Fig. 5 was applied. In order to find the mean values and variance/covariance matrices, the Expectation Maximization algorithm was used starting with 32 functions, as an example, Fig. 7 shows in the feature domain (after principal components analysis) five classes, corresponding to 5 fonts type. After have been found mean values and variance/covariance matrices, a Gaussian classifier was used to classifier any text having the fonts used in the learning phase. Table I shows the obtained confusion matrix. It can be observed that 8 types of fonts are 100% identified. Also, four different styles were 100% identified, as shown in table II. These results can be compared with those given in reference [9], where an average recognition of 99.1% is obtained with maximum and minimum intervals of recognition of 97.2% for Regular Arial and 100% for Times New Roman. It must be emphasized that 100% recognition of the type of font and style is achieved with the method proposed in this project. An average recognition value of 96.91% is obtained in reference [5] using a typographic approach, a value which is considerably lower than the obtained in this project.

Fig. 5. Analysis of estimation convergence (window size estimation): Third order moments $M_3(0,0)$ and Fourth order moments $M_4(0,0,0)$

Fig. 6. Maximum likelihood estimation by EM-Algorithm, this example shows only 5 different classes

Fig. 7. Fonts used during the recognitions (in regular style), (a)Arial, (b)Bookman, (c)Century Gothic, (d) Courier, (e)Comic Sans MS, (f)Impact, (g)Modern y (h)Times New Roman

Fig. 8. Of four styles, Times New Roman font, (a) regular, (b) bold, (c) italic y (d) italic -bold

Table 1. Confusion matrix for eight fonts in regular style, using a cross-validation method (50%) and a Bayes classifier

	Arial	Bookman	Century Gothic	Courier	Comic San MS	Impact	Modern	New Times Roman
Arial	100	0	0	0	0	0	0	0
Bookman	0	100	0	0	0	0	0	0
Century Gothic	0	0	100	0	0	0	0	0
Courier	0	0	0	100	0	0	0	0
Comic San MS	0	0	0	0	100	0	0	0
Impact	0	0	0	0	0	100	0	0
Modern	0	0	0	0	0	0	100	0
New Times Roman	0	0	0	0	0	0	0	100

Concerning the style recognition, it was recognized at 100%.

6 Conclusions and Future Works

A new approach was proposed and tested for font clustering. The method was based on the use of high-order central moments (third and fourth order), using a stochastic version of EM-algorithm in order to determine the number of different fonts and the function parameters (mean values and variance/covariance matrices). For classification task, a standard classifier (Bayes) was used. Non-isotropic characteristics were taken into consideration by making estimations in four different orientations. A principal component analysis reduces the number of features and the linear redundancies between them. Thus, results show a 100% performance, that is, a 0% error. The method worked well when differentiating types of fonts and styles. However, the method cannot identify the size of a letter, since the pre-processing stage of the method homogenizes every character to the same pre-established size. Future work involves the development of a technique where the size of fonts can be distinguished. This goal can be achieved by combining the text-independent technique proposed in this project with typographic and local approaches.

Acknowledgment

This work was partially supported by The Mexican National Council of Science and Technology (CONACYT) under project J31916.

References

1. G. Nagy, "Twenty Years of Document Image Analysis in PAMI", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 38-62, Jan 2000.
2. S. Khoubyari and J.J. Hull, "Font and function word identification in Document Recognition", *Computer Vision and Image Understanding*, vol. 63, no. 1, pp. 66-74, 1996.

3. R. Cooperman, "Producing Good Font Attribute Determination Using Error-Prone Information", *Int. Society for Optical Eng. J.*, vol. 3027, pp. 50-57, 1997.
4. H. Shi and T. Pavlidis, "Font Recognition and Contextual Processing for More Accurate Text Recognition", *Proc. Fourth Int. Conference Document Analysis and Recognition, (ICDAR'97)* pp. 39-44, Aug 1997.
5. A. Zramdini and R. Ingold, "Optical Font Recognition Using Typographical Features", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 877-882, Aug. 1998.
6. A. Schreyer, P. Suda, and G. Maderlechner, "Font Style Detection in Document Using Textons", *Proc. Third Document Analysis Systems Work-Shop, Assoc. for Pattern Recognition Int.*, 1998.
7. B. Julesz and J. R. Bergen, "Textons, the Fundamental Elements in Preattentive Vision and Perception of Textures", *The Bell System Technical J.*, Vol. 62, no. 6, pp. 1619-1645, July/August 1983.
8. J. Malik, S. Belongie, J. Shi and T. Leung, "Textons, Contours and Regions: Cue Integration in Image Segmentation", *IEEE International Conference on Computer Vision, Corfu, Greece, September 1999*.
9. Yong Zhu, Tieniu Tan and Yunhong Wang, "Font Recognition Based on Global Texture Analysis", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 10, pp. 1192-1200, October 2001.
10. Fukunaga K., "Introduction to Statistical Pattern Recognition", Academic press 1990. United Kingdom, 591 páginas.
11. A.C. Bovik, M. Clark, and W.B. Geisler, "Multichannel Texture Analysis Using Localized Spatial Filters", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 55-73, Jan 1990.
12. Haralick R. M., "Texture Feature for Image Classification", *IEEE Trans. On SMC*, 3(1):610-621, 1973.
13. Haralick R. M., "Statistical and Structural approaches to texture", *IEEE TPAMI* 67(5):786-804, 1979.
14. David H. Evans, "Probability and its Applications for Engineers", Prentice Hall, 2000, ISBN/Part Number: 0824786564.
15. Jain A. K. et al. "Statistical Pattern Recognition: a Review", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 1, January 2000, 4-37.
16. Avilés-Cruz, "Texture recognition by high order statistics: characterization and performance", PhD Thesis, Institute National Polytechnic of Grenoble L.T.I.R.F. Grenoble, France. 1997.
17. Avilés-Cruz, "Unsupervised texture segmentation using stochastic version of the EM algorithm and data fusion", *Proceedings International Conference on Pattern Recognition ICPR 98*, Vol. 2, pp. 1005-1009, Brisbane-Australia, August 1998.
18. T. Reed and J.M. Hans De Buf, "A Review of Recent Texture Segmentation and Feature Extraction Techniques", *CVGIP: Image understanding*, vol. 57, pp. 359-372, 1993.
19. Iftekharuddin K. M., Jemili K and Karim, M.A., "A feature-based neural wavelet optical character recognition system", *Aerospace and Electronics Conference*, 1995. NAECON 1995., Volume: 2, pp 621-628.
20. González R., "Digital Image Processing", Prentice Hall 1990.
21. G. S. Peake and T. N. Tan, "Script and Language Identification from Document Images", *Proc. BMVC'97*, vol. 2, pp. 169-184, Sept. 1997.

22. T.N. Tan, "Rotation Invariant Texture Features and their Use in Automatic Script Identification", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 7, pp. 751-756, July 1998.
23. G. R. Cross and A. K. Jain, 1983. Markov random field texture models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI vol. 5 pp. 25-39.
24. R. M. Haralick, K. Shanmugam, and I. Dinstein. 1973. Textural features for image classification. *IEEE Trans. Systems, Man, and Cybernetics*, vol. 3, no.6, pp. 610-621.
25. B. Julesz. 1975 Experiments in the visual perception of texture. *Scientific American*, no. 232, pp. 34- 43.
26. K.L Laws, 1980 Rapid texture identification, *SPIE: Image processing for missile guidance*, vol. 238, pp. 376-380.
27. T. Leung and J. Malik. 1999 Recognizing surfaces using three-dimensional textons. *IEEE International Conference on Computer Vision*, pp. 1010-1017.
28. H. Tamura, S. Mori, and T. Yamawaki, 1978, Textural features corresponding to visual perception, *IEEE Trans. on Systems., Man and Cybernetics*, SMC - 8, no. 6, pp. 460 – 472.
29. Tomita, Y. Shirai and S. Tsuji, 1982, Description of textures by a structural analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 4, no. 2, pp. 183-191.
30. L. Wang and D.C. He, 1990, Texture classification using texture spectrum, *Pattern Recognition*, vol. 23, pp. 905-910, 1990.
31. Zucker, S. W. and D. Terzopoulos, 1980, Finding Structure in Co-Occurrence Matrices for Texture Analysis, *Computer Graphics and Image Processing*, vol 2, pp 286-308.
32. L. Xu and M. I. Jordan, On Convergence Properties of the EM Algorithm for Gaussian Mixtures. C.B.C.L. paper No. 111, A.I. Memo No. 1520, Massachusetts Institute of Technology, July 1995.
33. J. Zhang and W. J. Modestino, Maximum Likelihood Parameter Estimation for Unsupervised Stochastic Model_Base Image Segmentation. *IEEE Trans. on Image Processing*, 3(4):404-420, July 1994.