

Character Prototyping in Document Images Using Gabor Filters

Bénédicte Allier and Hubert Emptoz

Reconnaissance des Formes et Vision Laboratory (RFV) – INSA Lyon
20 av. A. Einstein, 69621 Villeurbanne cedex, FRANCE
{allier;emptoz}@rfv.insa-lyon.fr

Abstract. In this article we present a particular application of Gabor filtering for machine-printed document image understanding. To do so, we assume that the text can be seen as texture, characters being the smallest texture elements, and we verify this hypothesis by a series of experiments over different sets of character images. We first apply a bank of 24 Gabor filters (4 frequencies and 6 orientations) on each set, then we extract texture features, that are used to classify character images without *a priori* knowledge using a Bayesian classifier. Results are shown for different characters written in a same font, and for different font types given a character.

1 Introduction

The advent of networks and computers, and the advantages they present regarding the manipulation and storage of written information, make the numerical support a more and more requested tool. Thus, the desire or the need for recovering always more information coming from paper documents, even old ones, supposes mechanisms of document recognition which are not limited to characters identification but which even require the analysis of its structure.

Technically, document image understanding involves document layout segmentation and functional (i.e. logical) labeling of the detected zones at different levels. The simplest one is text/non-text identification, but the knowledge of finer information such as font types and sizes should be useful to further analyze document images e.g. document image indexing and retrieval, document type classification... To achieve this, many methods have yet been proposed in the literature most of which work on segmented document images at a pixel level [1-5], thus suffering from noise images may present.

Our aim is then to elaborate a robust tool for the detection of character shapes and fonts (without explicitly recognizing the font type), working at a higher perceptual level, seeing machine-printed text as texture (characters being the smallest texture elements). In this way, we pay attention to general methods, classically used in natural images processing but still marginal for document images [6-8], and more precisely to Gabor filtering since it has been designed to simulate the human visual system [9].

Our paper is organized as follows. In the next section we will introduce the texture features we use, based on the well-known multi-channel Gabor filtering method.

Section 3, justifies that character prototyping can be seen as a textural problem. Results for font type prototyping are presented and discussed in section 4. Finally, section 5 presents a brief summary of the paper and remarks on our future work.

2 Texture Features Extraction and Gabor Wavelets

Multi-channel filtering techniques, inspired from the human vision system [9], enable the extraction of texture characteristics locally in frequency and in orientation. This is particularly the case of the Gabor theory presented below.

A 2D Gabor function h is a sinusoid plane wave modulated by a Gaussian envelope and oriented with angle θ from the x-axis. The mathematical formulation, in the spatial domain, for fundamental frequency u_0 along the x-axis (i.e. $\theta=0^\circ$), is:

$$h(x,y) = \underbrace{\exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right]}_{\text{Gaussian envelope}} \underbrace{\cos(2\pi u_0 x)}_{\text{sinusoidal function}} \quad (1)$$

where σ_x (resp. σ_y) is the Gaussian standard deviation along the x-axis (resp. the y-axis). Filters with orientation θ ($\theta \neq 0^\circ$) are obtained by rotating the latter equation.

In this approach, each filter is applied to the original textured image, and a further analysis enables the creation of a simple characterizing feature vector based on statistical calculations.

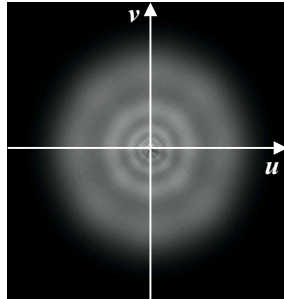


Fig. 1. Frequency representation of a bank of 24 Gabor filters: 6 frequency ranges, 4 orientations.

In our experiments, we use a bank of 24 Gabor filters, as presented in [10], defined in a 128×128 window, with $S = 4$ radial frequencies ($u_0 = 0.05, 0.1, \dots, 0.4$) and $K = 6$ values of orientation θ ($\theta = 0, \pi/6, \dots, 5\pi/6$) – cf. Fig. 1, since it seems to well characterize a wide range of textures. Also, we work on sets of input images composed of M prototype images on the one hand, and of N test images (that have to be classified) on the other hand. Original prototypes are character images randomly chosen among each class. Our method can then be divided into 3 main steps (cf. Fig. 2):

- (a) Gabor filters are applied to each input image independently, resulting in 24 filtered images per entry;
- (b) statistical features are calculated on each filtered image, producing as many feature vectors that enable the calculation of a distance measure between prototype and test images (as explained in the following paragraph);
- (c) distances are carried to a M -dimension space, on which a Bayesian classifier is processed, and final prototypes are elaborated superimposing all the test images composing each resulting class.

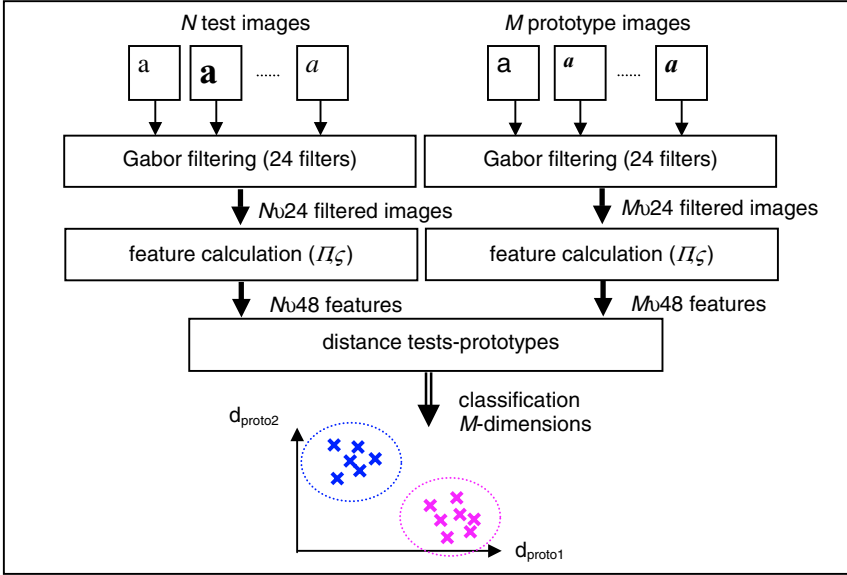


Fig. 2. Synopsis of the prototyping method using Gabor filters.

As we said before, our aim is to classify character images thanks to a distance measure calculated on feature vectors. For that, feature vectors we use for any input image are composed of statistical observations on its 24 related filtered images that characterize intensity repartition using first and second moments notated μ and σ . The distance measure is then defined on these features according to [10]. That is, for two response images i and j of features $(\mu_i^k; \sigma_i^k)$ and $(\mu_j^k; \sigma_j^k)$ over the k th filter, distance d_{ij} is given by:

$$d_{ij} = d(i; j) = \sum_{k=1}^{24} d_{ij}^k = \sum_{k=1}^{24} \left(\left| \frac{\mu_i^k - \mu_j^k}{\text{norm}(\mu)} \right| + \left| \frac{\sigma_i^k - \sigma_j^k}{\text{norm}(\sigma)} \right| \right)$$

where $\text{norm}(\mu)$ (resp. $\text{norm}(\sigma)$) are the standard deviations of μ (resp. σ) over the entire set of input images, used to normalize the two features.

3 Character Prototyping

The image set we are testing our method on, is composed of strongly structured machine-printed documents, presenting a few well distinguished font types. More precisely, our corpus is composed of 4 subsets of document images with homogeneous layouts, presenting in average 5 font types each. In our approach, text zones are considered as juxtapositions of small textured regions at a character level, so that different characters given a font type and a same character written in different font types should produce different textures e.g. ‘o’ and ‘z’ or ‘a’ and ‘a’ don’t have the same orientations nor the same apparent frequencies (italics are strongly inclined on the right, whereas classical font types are straight). This is what we want to show, testing our feature vector discriminant power over different character shapes.

Note that in all our experiments, we will be working on sets of binary characters (tests and prototypes) embedded in 128×128 images, coming from a preceding physical segmentation step (that aims at segmenting the character shapes in the document image, but which is not our purpose here), randomly chosen among binarized document images of the corpus.

3.1 Tests

In the first series of experiments, the chosen characters are the 14 most frequent ones in our documents, all written in the same font: ‘a’, ‘c’, ‘d’, ‘e’, ‘i’, ‘l’, ‘m’, ‘n’, ‘o’, ‘p’, ‘r’, ‘s’, ‘t’, ‘u’ (Fig. 3.a), resulting in ≈ 560 character images. Feature vectors, and distances are calculated between test images and prototype images, and carried on a 14-dimension space for classification purposes. The classification is processed using AutoClass software, presented in [11], involving the Bayesian classification theory and aiming at automatically determining the maximally probable number of classes with the classical finite mixture distribution as fundamental model.

3.2 Results and Discussion

The classification algorithm is processed 500 times on the same dataset, since the search includes a random component, and we obtain 17 classes (corresponding extracted prototypes are showed on Fig. 3.b). Examining the results, we notice that the classifier has managed to separate well very particular patterns: ‘e’, ‘r’, ‘s’, ‘t’, ‘i’, ‘a’, ‘o’, ‘l’, ‘m’ and ‘c’ (i.e. classes #0, 2 to 8, 10 and 11), producing character prototypes compliant to the original ones. 3 other classes are composed of mixed characters: one composed of ‘n’ and ‘u’, resulting in the prototype #1 on Fig. 3.b of shape ‘n’ (*ns* are more numerous than *us*), and two composed of ‘d’ and ‘p’, resulting in prototypes #9 and #12 of shape ‘d’ (*ds* are more numerous than *ps*). These are not explicitly classification errors since Gabor filtering has been designed to be rotation and translation invariant, and since ‘n’ and ‘u’ are 2 iterations of a same pattern rotated with angle 180° . Classes #13 to 17 are rejection classes, containing on the most 5 test images each. Finally, the error rate is evaluated to $\approx 3.6\%$, i.e. we get a 96.4% recognition rate, which is really satisfying. These results show that Gabor filtering is

useful at a character level for prototype extraction, except rotation problems that have to be solved now.

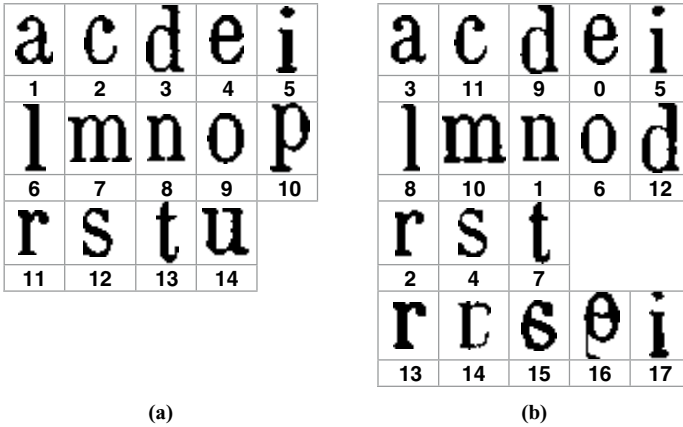


Fig. 3. (a) Character images given a font type, randomly chosen as prototype with corresponding class numbers and (b) resulting prototypes and class numbers using Gabor filters.

Our idea is to complete our characterization process using a very simple non time-consuming distance measure. For that, we use a pattern matching method presented in [12], inspired from [13], based on the calculation of a symmetrical difference between a reference and a test pattern using masks. Thus, we get a 92.4% recognition rate for characters ‘u’ and ‘n’ over 79 samples, and a 100% recognition rate for characters ‘d’ and ‘p’ over 42 samples. Resulting prototypes are shown on Fig. 4. These results are encouraging since they can be obtained using a very simple algorithm.



Fig. 4. Character prototypes extracted using Gabor filtering coupled to our simple pattern matching method.

Another interesting study (involving the same classification process) has been made to evaluate the discriminant power of the character shapes according to Gabor analysis. The idea was to use a restricted number of prototypes (beginning with 2 and increasing this number progressively) to classify the ≈ 560 test images over 16 patterns: ‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘h’, ‘i’, ‘l’, ‘m’, ‘n’, ‘o’, ‘p’, ‘r’, ‘s’, ‘t’, ‘u’. In the first experiment, we began with character prototypes ‘o’ and ‘z’ since they seem to be the most particular ones. This gave only 6 classes, and among them only 2 well distinguished shapes: ‘r’ and ‘o’ (e.g. one class confounds *as*, *cs*, *es* and *ns*). Thus, we introduced a third prototype ‘a’ since it is hardly classified in the preceding test and since it seems to be a very singular shape. This gave 10 classes among which 5 patterns were correctly identified (‘n’ and ‘u’, ‘r’, ‘s’, ‘a’ and ‘o’), and one rejection class (composed of 3 characters). The fourth prototype introduced was character ‘t’,

since our process didn't separate vertical patterns and since it was still very different from the three other prototypes. This gave 13 classes in which almost only *ls* and *ts* were hardly discriminated. The calculation of an error rate was possible here: it reaches 3.54% i.e. the recognition rate is more than 96.4%. Adding a last prototype 'l' provides an error rate of 2.47%, that is the recognition rate is evaluated to more than 97.5%. In other words, with Gabor analysis, 4 character prototypes are enough to classify 16 character shapes with an error rate of only 3.54%.

Finally, the experiments we made prove that Gabor filtering is useful at a character level for characterization, thus enabling character prototype extraction.

4 Font Prototyping

We carry on studying this tool for font type prototyping (we are not trying to give them a name). For that, we are interested in a single pattern written in different fonts.

4.1 Tests

The method used in this experiment is the same as before. The character we chose is 'a' since it is widely used in French, and since it is present even in poorly used font types in our documents. Randomly choosing character samples in our dataset results in 11 font types over | 370 character images (cf. Fig. 5.a). As before, we use the AutoClass algorithm, in an 11-dimension space.

4.2 Results

The classification algorithm is again processed 500 times on the same dataset. We obtain 13 classes, which extracted prototypes are shown on Fig. 5.b. Among these classes 11 contain single font types (classes #0 to 9 and 11 on Fig. 5.b), even though some font types look very similar (e.g. original classes #2 and 5), and 2 are rejection classes representing all in all 8 samples. Note that there is no confusion between original classes #1 and 9 which have only a small difference in the upper tail of the character. Finally, the error rate is evaluated to $\approx 3.0\%$, that is we get $\approx 97.0\%$ recognition rate, which is really interesting considering the low representativity of some classes.

Experimentations have also been made on character 's' over ≈ 350 test images, resulting in a 96.85% recognition rate.

Finally, Gabor filtering seems also to be a good tool for character font type prototyping, even if we are conscious that further experiments should be done on other character shapes to enforce our diagnostic.

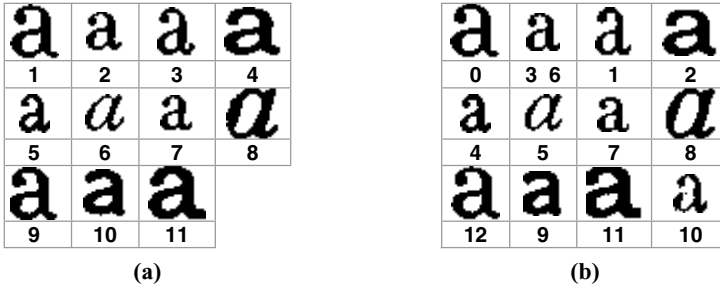


Fig. 5. (a) Character images given a font type, randomly chosen as prototype with corresponding class numbers and (b) resulting prototypes and class numbers using Gabor filters.

5 Discussion and Further Work

We have presented an interesting approach for character and font type prototyping based on a global textural approach using Gabor filtering. We have shown that a bank of 24 Gabor filters (4 frequencies and 6 orientations) was enough to discriminate 11 font types and 14 character shapes. As Gabor filters were designed to be neither translation nor rotation sensitive, we had to complete our method with a simple shape matching algorithm to distinguish between similar characters (from Gabor point of view) such as ‘u’ and ‘n’ or ‘d’ and ‘p’. Nevertheless, we achieve a 96.8% global average recognition rate over the two experiments.

Globally, results are encouraging since they allow the grouping of binarized randomly chosen characters, sometimes suffering from strong artifacts that algorithms at a pixel level (like the pattern matching solution of section 3) would have considered as different (a complete analysis is presented in [12]). More experiments should be done on character shapes: this will be part of our future work.

Our method doesn’t allow the recognition of font attributes separately yet (e.g. font type or name (Courier, Helvetica...), font size...), this should be done in a further analysis. Such information is useful in practice since it enables document structure extraction by ways of Intelligent Character Recognition (ICR) in real world [14].

The results obtained are strongly influenced by the physical segmentation step, which is aimed at isolating the characters. Automating the method needs the use of a pre-processing phase as perfect as possible, which is hardly reachable. In real world applications, users should be able to operate on the system at any time. Moreover, the discriminative post-processing step should be enforced by listing every possible confusion between patterns from Gabor point of view (e.g. ‘n’ and ‘u’ or ‘d’ and ‘p’ as seen before, but also not seen intuited cases like ‘b’ and ‘q’).

Finally, we should focus on the possibility of lowering the number of characterizing Gabor filters and features, using the method proposed in [15], to avoid useless computational burdens. This work is being processed at this very moment.

References

1. Chaudhuri, B. B. et Garain, U. Extraction of type style-based meta-information from imaged documents. *IJDAR*. Vol. 3 (3) (2001) 138-149.
2. Wong, K. Y., Casey, R. G. et Wahl, F. M. Document Analysis System. *IBM Journal of Research and Development*. Vol. 26 (6) (1982) 647-656.
3. Eglin, V., Bres, S. et Emptoz, H. Statistical characterization and classification of printed text in a multiscale context. *Proc. of 2nd Int. Workshop on SPR, Sydney (NSW), (Australia), (1998) 960-967.*
4. Duffy, L. Recherche d'information logique dans les documents à typographie riche et récurrente. Application aux sommaires. Thesis of the INSA de Lyon, Lyon (France) (1997).
5. Doermann, D. S., Rivlin, E. et Rosenfeld, A. The function of documents. *IJCV*. Vol. 16 (11) (1998) 799-814.
6. Jain, A. K., Bhattacharjee, S. K. et Chen, Y. On texture in document images. *Proc. of IEEE CVPR, Champaign, Illinois (USA), (1992) 677-680.*
7. Wu, V., Manmatha, R. et Riseman, E. M. Finding Text in Images. *Proc. of ACM Int. Conf. on Digital Libraries, Philadelphia, PA (USA), (1997) 23-26.*
8. Zhu, Y., Tan, T. et Wang, Y. Font Recognition Based On Global Texture Analysis. *IEEE Trans. on PAMI*. Vol. 23 (10) (2001) 1192-1200.
9. Porat, M. et Zeevi, Y. Y. The generalized Gabor scheme of image representation in biological and machine vision. *IEEE Trans. on PAMI*. Vol. 10 (4) (1988) 452-468.
10. Manjunath, B. S. et Ma, W. Y. Texture Features for Browsing and Retrieval of Image Data. *IEEE Trans. on PAMI*. Vol. 18 (8) (1996) 837-842.
11. Cheeseman, P. et Stutz, J. Bayesian classification (Autoclass): Theory and results. *Adv. In Knowledge Discovery And Data Mining (1996) 153-180.*
12. Allier, B. et Boursier, N. Reconstruction de caractères dégradés: étude pour la détermination d'un caractère "idéal". Lyon (FRANCE): Institut National des Sciences Appliquées - INSA; Report No.: RR2002-05 (décembre 2002).
13. Chenevoy, Y. Reconnaissance structurelle de documents imprimés : études et réalisations. Thesis of the INPL, (1992).
14. Zramdini, A. Study of Optical Font Recognition Based on Global Typographical Features. Thesis of the Université de Fribourg (Suisse) (1995).
15. Jain, A. K. et Farrokhnia, F. Unsupervised Texture Segmentation Using Gabor Filters. *Pattern Recognition*. Vol. 24 (12) (1991) 1167-1186.