

Efficient distance-based per-pixel texture classification with Gabor wavelet filters

Jaime Melendez · Miguel Angel Garcia ·
Domenec Puig

Received: 18 December 2006 / Accepted: 19 October 2007 / Published online: 4 December 2007
© Springer-Verlag London Limited 2007

Abstract This paper proposes an efficient solution to the problem of per-pixel classification of textured images with multichannel Gabor wavelet filters based on a selection scheme that automatically determines a subset of prototypes that characterize each texture class. Results with Brodatz compositions and outdoor images, and comparisons with alternative classification techniques are presented.

Keywords Texture · Per-pixel texture classification · Non-parametric distance-based classification · Multichannel Gabor wavelet filters · Prototype selection

1 Introduction

Texture is a major visual cue utilized in a wide variety of computer vision tasks as it allows distinguishing among different objects or surfaces that have similar shape or

color. Unfortunately, it is the visual cue that is most difficult to model, as it is intrinsically noisy by nature and affected by various external factors, such as illumination, rotation, and scale, which alter its perception. This complexity has fostered a large amount of research during the last decades.

Two of the main approaches to texture-based image analysis are: *unsupervised texture segmentation*, which consists of partitioning a given image into regions of uniform texture, although without recognizing the texture class associated with every region, and *supervised texture classification*, whose goal is to identify all or some of the texture patterns present in the input image given a set of known classes of interest. In particular, *supervised per-pixel* (also known as *pixel-based*) texture classification aims at recognizing the texture patterns to which the pixels of a given image belong. In order to accomplish this task, it is necessary to compute a set of texture features by evaluating one or more texture feature extraction methods in a neighborhood of every pixel. This neighborhood is usually defined as a square window centered at that pixel.

Regarding texture classification, considerable effort has been devoted to the classification of single-textured images (e.g., [1–4]). In this problem, known as *whole-image* (or *full-image*) classification, every texture pattern of interest is usually characterized by computing a single feature vector (prototype). Another feature vector is then obtained in a similar way from the given input image to be recognized. Finally, this image is classified into the texture pattern whose prototype is closest to the image's feature vector.

However, single-textured image classifiers cannot be directly applicable to the problem of per-pixel classification. To start with, in this problem it is necessary to compute a single feature vector per pixel, not per image. This implies that those vectors must be determined based

This work has been partially supported by the Spanish Ministry of Education and Science under projects DPI2004-07993-C03-03 and DPI2007-66556-C03-03.

J. Melendez (✉) · D. Puig
Intelligent Robotics and Computer Vision Group,
Department of Computer Science and Mathematics,
Rovira i Virgili University, Av. Països Catalans 26,
43007 Tarragona, Spain
e-mail: jaime.melendez@urv.cat

D. Puig
e-mail: domenec.puig@urv.cat

M. A. Garcia
Department of Informatics Engineering,
Autonomous University of Madrid, Francisco Tomas y Valiente 11,
28049 Madrid, Spain
e-mail: miguelangel.garcia@uam.es

on the information contained in the neighborhood of every pixel. This process must be carried out over both the input image to be classified and the training images corresponding to the different texture patterns. Regarding the latter, this poses the problem of deciding how many prototypes are utilized to characterize each texture pattern, as every training image potentially generates as many prototypes as pixels. The use of so many prototypes can be prohibitive in real applications in terms of both computational time and memory usage.

The present paper provides a solution to the aforementioned problem by automatically selecting a subset of prototypes per texture pattern. Those prototypes are obtained after applying a modified k -means clustering algorithm to the feature vectors extracted from the available training images. Afterwards, a K -nearest neighbors classifier (K -NN) is used to finally assign a label to every pixel of the classified image. The parameters of both the proposed variation of k -means and K -NN are automatically selected for every texture pattern. Feature vectors are obtained by processing the images with a multichannel Gabor wavelet filter bank by following the per-pixel methodology described above. Multisized windows are also utilized in order to improve the accuracy of the classifier near boundaries between regions of different texture. The use of multiple window sizes has already proven to be advantageous for per-pixel texture classification [5, 6]. The proposed technique is effective in terms of classification rates, memory consumption, and computation time.

This paper is organized as follows. Section 2 gives a detailed explanation of the stages of the proposed per-pixel texture classifier. Section 3 describes the technique for automatic parameter selection. Section 4 shows experimental results. Finally, conclusions and future work are given in Sect. 5.

2 Proposed Gabor-based, per-pixel classifier

The texture classification methodology proposed in this work is as follows. During an initial training stage, a set of prototypes is extracted from every texture pattern. The training image/s associated with each pattern are first filtered by applying a multichannel Gabor filter bank, obtaining a cloud of texture feature vectors for every pattern. A set of prototypes is then extracted in order to represent that cloud.

During the evaluation stage of the classifier, a given test image is processed in order to identify the texture pattern corresponding to each of its pixels. This is done by first applying the multichannel Gabor filter bank to the test image. A feature vector is thus obtained for every pixel. Each vector is classified into one of the given texture

patterns by a K -NN classifier fed with the prototypes extracted during the training stage. The final classified image is obtained after post-processing that result. The stages involved in this scheme are detailed below.

2.1 Extraction of texture feature vectors

A wide variety of texture feature extraction methods have been proposed in the literature. Among them, multichannel filtering techniques based on Gabor filters have received considerable attention [7]. A multichannel Gabor wavelet filtering approach has been used for the texture feature extraction stage of this work. In particular, the design originally proposed in [1] is followed.

The texture features that characterize every pixel and its surrounding neighborhood (window) are both the mean and standard deviation of the module of the Gabor wavelet coefficients. The Gabor filter bank has been configured with six scales and four orientations according to [8], and a range of frequencies between 0.05 and 0.4 according to [1]. Therefore, every feature vector has a total of 6 (no. of *scales*) \times 4 (no. of *orientations*) \times 2 (*mean* and *SD*) = 48 dimensions. All dimensions are normalized between 0 and 1.

The means and deviations mentioned above are computed for W different window sizes (W is set to 6 in this case): 3×3 , 5×5 , 9×9 , 17×17 , 33×33 and 65×65 , according to [5]. Thus, W sets of feature vectors are generated for each of the given texture patterns during the training stage, as well as for the test image during the evaluation stage. The Gabor filter's kernel size has been set to coincide with the window size.

2.2 Extraction of prototypes from feature vectors

The goal of this stage is to obtain a reduced set of prototypes that represent a given set of 48-dimensional feature vectors. A variation of the k -means clustering algorithm is applied, in such a way that the number k of clusters is not predefined, but derived from a resolution parameter R that determines the size of those clusters. R is defined as a fraction of the longest diagonal L of the bounding box that delimits the whole feature space.

The proposed clustering algorithm has two main stages: splitting and refinement. Regarding the splitting stage, let us suppose that the available feature vectors have been split into C disjoint clusters (initially $C = 1$). The bounding box that delimits the volume of each cluster is found. If the length of its longest diagonal is greater than $R \times L$, the cluster is split into two. Hence, the lower the value of R , the larger the number of clusters. R is selected as described in Sect. 3. Therefore, after a single pass of the algorithm,

there will be between C (if no clusters are split) and $2C$ (in case all previous clusters are split) clusters.

The splitting of a cluster is done by dividing the bounding box of the original cluster across its longest dimension. Every resulting cluster is associated with both a bounding box and its corresponding centroid. After the splitting stage, the refinement stage consists of applying the k -means algorithm, but using all the available centroids as initial seeds. Both the splitting and refinement stages are repeated until no new clusters are generated.

2.3 K -NN and post-processing

The K -NN classifier is implemented without variations. It classifies a given texture feature vector into one of the T texture patterns of interest. For every pattern, it considers a set of prototypes computed as described in Sect. 2.2. The value of parameter K is determined as described in Sect. 3. A final post-processing stage aims at removing the noisy regions that usually appear after applying the K -NN classifier.

3 Automatic parameter selection

The classifier described in Sect. 2 depends on two parameters: the resolution R that limits the clusters' size, and the number of neighbors K of K -NN. The performance of the classifier greatly depends on the choice of those parameters. A simple parameter selection algorithm entirely based on the training set is also applied during the training stage.

The selection algorithm applies the proposed classifier in order to compute classification rates for the training images corresponding to the T texture patterns of interest.

The classifier is run with different combinations of R and K . Since an exhaustive search for R and K is prohibitive, a sampled search is performed with R varying from 1.0 to 0.3 with decrements of 0.1 (8 different values), and K being 2^n , where n varies from 0 to 8 with increments of 1 (nine different values). However, some of the 8×9 combinations of those parameters are unfeasible, since some values of K are not valid depending on the value of R as discussed below.

Given a certain value of R , let p be the number of prototypes of the texture pattern with the minimum number of clusters according to the prototype extraction algorithm described in Sect. 2.2. In order to guarantee that all texture patterns can be identified, K has been forced to be lower than or equal to p . Without this constraint, some texture patterns could not be identified, as it is possible to reach a point where all the p available votes for a texture pattern have been used (the available votes for a texture pattern coincide with the number of modeling prototypes considering all window sizes); but as K is greater than p , the remaining $K-p$ votes will only consider the textures that can still vote. Thus, in the experiments conducted in this work, 53 different combinations of R and K have been considered.

Next, the average of all classification rates for each combination of R and K by considering all the given texture patterns is computed. For instance, Fig. 1 shows the average classification rates for the texture patterns belonging to the first test image in Fig. 2. The best combination of R and K is marked with a circle. In general, the finer the resolution R , the larger the number of neighbors K necessary to achieve good classification results.

In the end, the combination of R and K that yields the maximum classification rate is selected as the configuration parameters of the classifier described in Sect. 2.

Fig. 1 Average classification rate curves for the patterns belonging to the first test image in Fig. 2 by considering different combinations of R and K . The best combination is marked with a circle

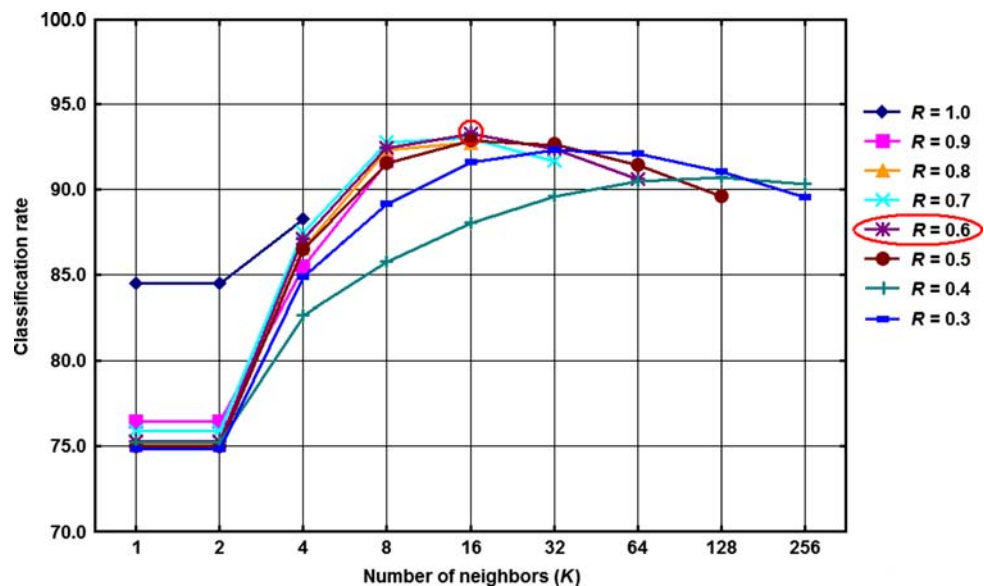
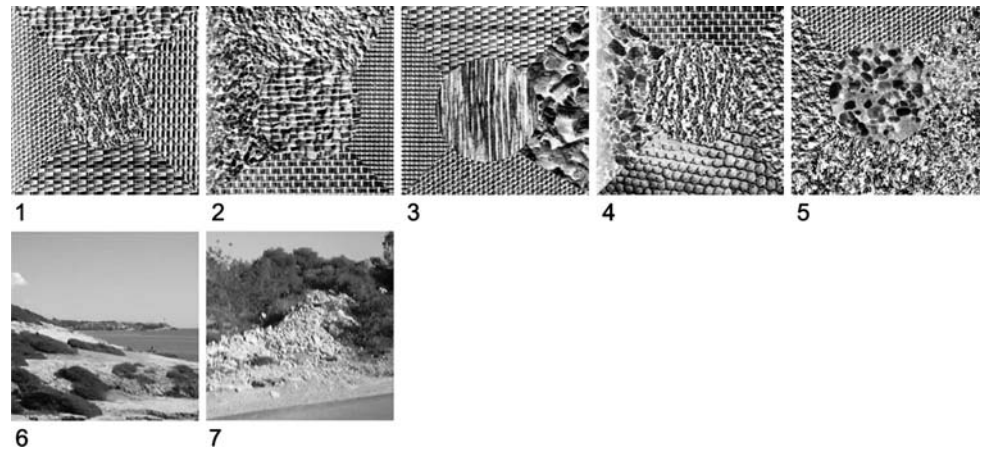


Fig. 2 Experimental test images: Brodatz compositions based on [9] and outdoor real images



4 Experimental results

The proposed technique has been evaluated on composite images based on those presented in [9] and widely used thereafter, which are made of patches of the well-known Brodatz textures [10], and on real outdoor images. Figure 2 shows seven of those test images. In both cases, image regions used for training are different than those used in the test stage.

First, in order to validate the proposed methodology for automatic parameter selection, the combination of R and K that yields the best classification results for the given test images is found. Then, the classification rates produced by that combination of parameters have been compared with the rates obtained with the parameters automatically chosen by the selection algorithm described in Sect. 3. All these classification rates have been obtained without considering the post-processing stage in Sect. 2.3. Table 1 summarizes the results of this comparison.

These results indicate that the proposed selection algorithm is effective in finding a reasonable pair of parameters R and K . Actually, if the classification rates corresponding to the 53 feasible combinations of those parameters are computed, the combination of R and K that is automatically chosen by the selection algorithm always scores within the ten best possible classification rates and it even coincides

with the best configuration in two cases as shown in the last column of Table 1.

Next, a comparison in terms of classification rates, memory consumption, and computation time has been performed, considering the two extreme alternatives that naturally fit in the approach of the proposed classifier, namely: the minimum distance classifier, which only uses both a single prototype per texture pattern and window size, and one nearest neighbor; and the classifier that stores all the available feature vectors as prototypes for the given training texture patterns. In the latter case, the window size that leads to the best classification rate is chosen in order to yield a comprehensive computation time.

Results are shown in Table 2. Figure 3 displays two of the classification maps produced by the three approaches. The first column shows the ground-truth for each test image. For the outdoor images (6 and 7), black areas represent image regions that do not correspond to any of the sought texture patterns. Pixels that belong to these “unknown” texture patterns have not been taken into account when computing classification rates. On the other hand, black borders appearing in the classification maps correspond to those pixels that could not be classified because of the minimum window size used. Again, these pixels have been discarded for the reported classification rates.

Table 1 Parameters and corresponding classification rates (%) for the classifier configured with both optimal parameters and the proposed parameter selection algorithm, considering the test images in Fig. 2

Image	Optimal parameters			Selected parameters			Classification rate rank
	R	K		R	K		
1	$R = 0.8$	$K = 16$	91.1	$R = 0.6$	$K = 16$	89.5	7/53
2	$R = 0.7$	$K = 32$	86.2	$R = 0.7$	$K = 32$	86.2	1/53
3	$R = 0.5$	$K = 32$	91.6	$R = 0.5$	$K = 16$	91.3	5/53
4	$R = 0.7$	$K = 32$	78.4	$R = 0.6$	$K = 32$	77.4	5/53
5	$R = 0.7$	$K = 16$	72.9	$R = 0.8$	$K = 16$	71.1	7/53
6	$R = 0.3$	$K = 4$	93.1	$R = 0.3$	$K = 4$	93.1	1/53
7	$R = 0.3$	$K = 16$	81.5	$R = 0.3$	$K = 4$	79.9	6/53

Table 2 Classification rates (%), computation time in seconds (Pentium 4 at 3.0 GHz), and number of selected prototypes for the proposed classifier and alternative approaches

Images	Minimum distance classifier			Proposed classifier			“All-prototypes” modeling classifier		
	Classification rate	CPU time	Number of prototypes	Classification rate	CPU time	Number of prototypes	Classification rate	CPU time	Number of prototypes
1	94.9	2	30	95.2	12	763	96.1	2,930	307×10^3
2	90.4	2	30	91.7	7	329	92.9	2,569	288×10^3
3	95.0	2	30	97.0	25	1,633	97.2	2,929	307×10^3
4	79.2	2	30	88.2	15	790	86.9	2,578	288×10^3
5	73.0	2	30	78.6	4	186	80.0	2,936	307×10^3
6	91.0	1.8	24	93.3	19	1,708	92.6	1,266	127×10^3
7	78.1	2	30	84.3	23	1,950	78.6	558	63×10^3

From Table 2, it is clear that the proposed classifier is a good trade-off between the two extreme approaches as, on average, the memory usage is around $220\times$ lower than the memory used when storing as many prototypes as pixels, and the computational time to classify an input image is $150\times$ lower. In turn, the degradation of the proposed classifier in terms of classification rates is small and below 3 units on average. Notice that, in some cases, the proposed classifier even yields slightly better results.

On the other hand, the minimum distance classifier is the fastest among the three compared alternatives as expected. In turn, the proposed classifier requires significantly more memory than the minimum distance classifier, although it can be easily handled by current computers. In terms of classification rates, however, the difference is notoriously in favor of the proposed classifier, especially in the fourth and fifth test images, which are considered among the most difficult ones as demonstrated by the low achieved scores.

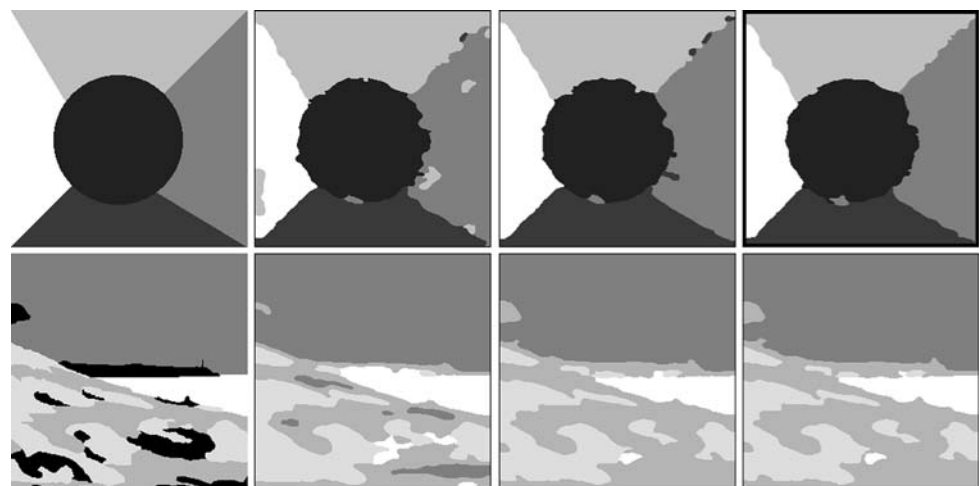
A comparison in terms of classification rates with other supervised texture classification techniques has also been performed. These techniques are: the texture classifier based on integration of multiple methods and windows

described in [5], and both the K -NN ($K = 5$) and multivariate Gaussian (MVG) classifiers provided with the MeasTex suite [11]. In order to achieve per-pixel classification with MeasTex, which is a framework oriented to the classification of complete images instead of individual pixels, the classifiers of MeasTex have been utilized to classify every pixel of the test images given a subimage of 32×32 pixels quasi-centered at that pixel (32×32 pixels is the minimum subimage size accepted by MeasTex).

The texture features and window sizes used by the integration strategy are the same as the ones discussed in Sect. 2.1 and will be referred to as *optimized Gabor features*. In turn, the texture features used by the K -NN and MVG classifiers are energy measures derived from the coefficients produced by the classical Gabor filter bank implementation also included in MeasTex. These features will be referred to as *non-optimized Gabor features*.

The results of this comparison are summarized in Fig. 4. Figure 5 shows two of the classification maps produced after the respective classifications. In all cases, the proposed classifier produced better classification rates than the other evaluated techniques.

Fig. 3 Classification maps for images 3 and 6 in Fig. 2 produced by the minimum distance classifier (*second column*), the proposed classifier (*third column*) and the classifier that stores all feature vectors as texture models (*fourth column*). Corresponding ground-truth (*first column*)



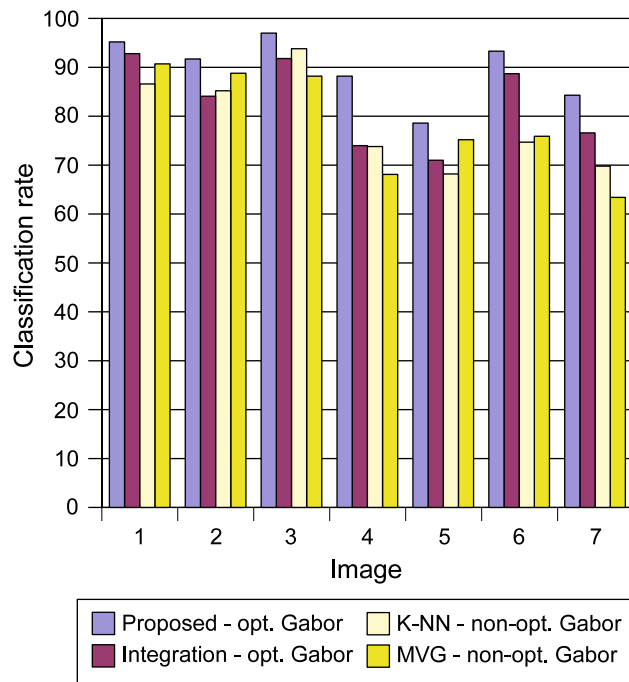


Fig. 4 Classification rates (%) for the compared classification techniques (proposed classifier, integration-based classifier, and MeasTex classifiers) using texture features derived from both optimized and non-optimized Gabor filters

Finally, in order to assess the performance of the proposed classification algorithm applied to optimized Gabor filters, another comparison with three alternative schemes has been conducted: (a) a straightforward extension to per-pixel classification of the LBP-based classifier proposed in [2], with its same *local binary pattern* (LBP) operators as feature extractors and the *G* statistic as dissimilarity measure. This extension has been performed in a similar way as the one described for MeasTex, but with subimages of 16×16 as suggested in [2], and keeping all feature vectors as texture models. (b) The *K*-NN ($K = 5$) classifier of MeasTex, and (c)

the MVG classifier of MeasTex. Both MeasTex classifiers have been independently evaluated with features derived from the fractal dimension, gray level co-occurrence matrices (GLCM) and Markov random fields (MRF).

The results presented in Fig. 6 show that the proposed combination of classifier and texture features yields significantly better classification rates than the other alternatives in the majority of cases. The same applies to the resulting classification maps. Two of them are shown in Fig. 7. The few exceptions to this statement occur in test image 5 (see Fig. 2) for the LBP features with the adapted *G* statistic classifier, and for the GLCM and MRF features with the MVG classifier.

Apparently in this case, the aforementioned texture features are able to better distinguish among the sought texture patterns than the optimized Gabor wavelet filter bank. In the case of the MRF features, this is specially true for the patches located both to the right and to the bottom of the composition. This situation is quite remarkable as this pair of texture patterns are perceptually very similar and difficult to segment even for a human observer.

5 Conclusions

This paper presents a new distance-based, per-pixel texture classifier based on clustering techniques and multichannel Gabor wavelet filters that achieves good classification results with low computational times and affordable memory usage. The proposed classifier has been compared against the simplest strategy, which stores a single prototype per texture pattern and window size, and applies a nearest neighbor classifier, as well as with the alternative extreme approach, which stores all the available texture feature vectors extracted from the training set. Comparisons with other recent and traditional supervised classification

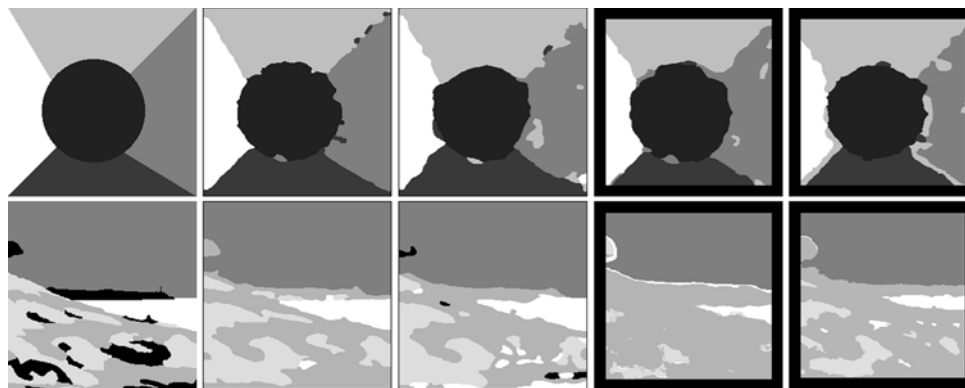


Fig. 5 Classification maps for images 3 and 6 in Fig. 2 produced by the proposed classifier (*second column*), the integration strategy in [5] (*third column*), the *K*-NN classifier included in MeasTex (*fourth column*) and the MVG classifier also included in MeasTex (*fifth*

column). Corresponding ground-truth (*first column*). The results for the first two classifiers consider optimized Gabor filters, while the ones for the last two classifiers consider the classical, non-optimized Gabor filters implemented in MeasTex

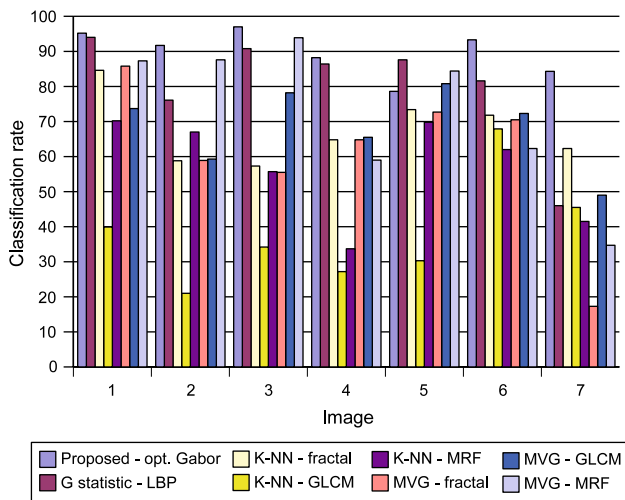


Fig. 6 Classification rates for different combinations of classification technique and texture feature extraction methods

schemes have been also carried out. In almost all cases, the proposed classifier yielded the best classification rates.

A selection technique for automatically choosing the configuration parameters corresponding to: (a) the resolution R , which determines the size of the clusters and, therefore, the number of seeds for k -means, and (b) the number of neighbors, K , used by the K -NN classifier has also been developed. Experiments have shown that this selection technique determines configuration parameters that yield classification rates very close to the optimal ones.

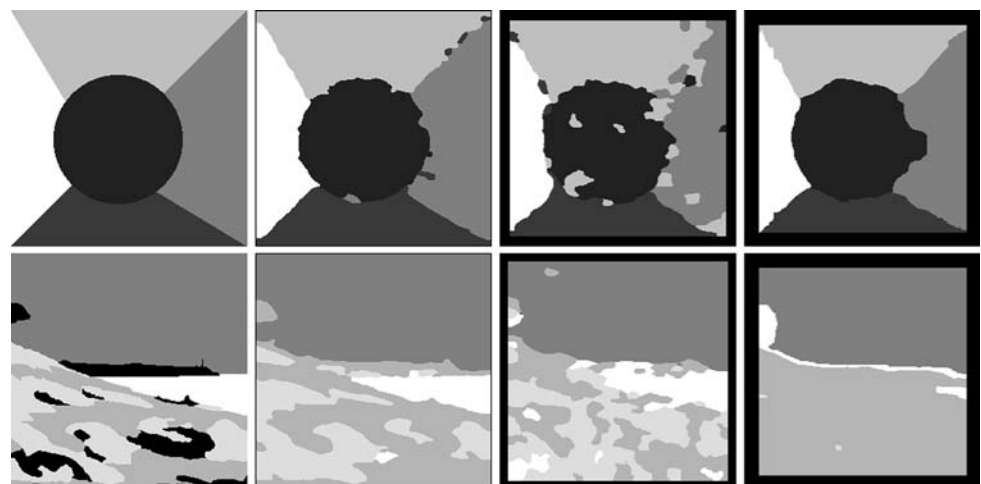
Further work will consist of studying better schemes for integration of different window sizes in order to reduce the information sources when classifying a given feature vector and thus improving the accuracy and computational time of the current algorithm. Additionally, as the proposed methodology is thought to be valid for any texture method or group of methods provided they generate feature vectors, we are planning to extend this technique in order to

integrate different feature extraction methods in a coherent way. Finally, we aim at adapting the proposed methodology to unsupervised per-pixel texture segmentation. The goal in this case is to automatically determine sets of prototypes that characterize the different regions of homogeneous texture within a given image.

6 Originality and contribution

This paper proposes a new, efficient, non-parametric, distance-based classifier as a solution to the per-pixel texture classification problem. Within this problem, every image pixel must be characterized by a feature vector in a local neighborhood, thus potentially generating an enormous number of prototypes to compare with during the classification stage. Using a resolution-driven clustering algorithm, which is a wrapper around the classical k -means clustering, the proposed technique selects a reduced subset of prototypes in order to model the different texture classes. Then, a K -nearest neighbors classifier is used to finally assign a label to every pixel in the input image. As the performance of the classifier greatly depends on both the resolution used for clustering and the number of neighbors used for classification, an automatic parameter selection algorithm that is able to find a suitable combination of those parameters entirely based on the training set, has been also proposed. Experiments with Brodatz compositions and outdoor images, and comparisons with alternative classification techniques show that the developed classifier is effective in terms of classification rates, memory consumption, and computation time. Applications that would potentially benefit from this technique are image analysis tasks, such as identification of specific types of soil in satellite or aerial images, detection of tumors in medical images, fabric defect detection, among others.

Fig. 7 Classification maps for images 3 and 6 in Fig. 2 produced by the proposed classifier with optimized Gabor filters (second column), an extension of the LBP-based classifier proposed in [2] (third column), and the best result among the tested MeasTex classifiers (fourth column). Corresponding ground-truth (first column)



References

1. Manjunath BS, Ma WY (1996) Texture features for browsing and image retrieval. *IEEE Trans Pattern Anal Mach Intell* 18(8):837–842
2. Ojala T, Pietikäinen M, Mäenpää T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans Pattern Anal Mach Intell* 24(7):971–987
3. Xu CL, Chen YQ (2006) Classifying image texture with statistical landscape features. *Pattern Anal Appl* 8(4):321–331
4. Photos VK, Theoharatos C, Zygouris E, Economou G (2007) Distributional-based texture classification using non-parametric statistics. *Pattern Anal Appl*. [10.1007/s10044-007-0083-9](https://doi.org/10.1007/s10044-007-0083-9)
5. Garcia MA, Puig D (2007) Supervised texture classification by integration of multiple texture methods and evaluation windows. *Image Vis Comput* 25(7):1091–1106
6. Puig D, Garcia MA (2006) Automatic texture feature selection for image pixel classification. *Pattern Recognit* 39(11):1996–2009
7. Grigorescu SE, Petkov N, Kruizinga P (2002) Comparison of texture features based on Gabor filters. *IEEE Trans Image Process* 11(10):1160–1167
8. Chen L, Lu G, Zhang D (2004) Effects of different Gabor filter parameters on image retrieval by texture. In: *Proceedings of the tenth international multimedia modelling conference*, pp 273–278
9. Randen T, Husoy JH (1999) Filtering for texture classification: a comparative study. *IEEE Trans Pattern Anal Mach Intell* 21(4):291–310
10. Brodatz P (1999) *Textures: a photographic album for artists and designers*. Dover & Gree Publishing Company
11. Smith G, Burns I (1997) Measuring texture classification algorithms. *Pattern Recognit Lett* 18(14):1495–1501

Author Biographies



Jaime Melendez received the M.S. in Electronics Engineering from Antenor Orrego University, Trujillo, Perú, in 2002, Information Technologies from Polytechnic University of Madrid, Madrid, Spain, in 2005, and Computer Engineering from Rovira i Virgili University, Tarragona, Spain, in 2007. He is currently a Ph.D. candidate at Polytechnic University of Catalonia and member of the Intelligent Robotics and Computer Vision Group at Rovira i Virgili

University. His research interests include pattern recognition, image processing, texture analysis and robotics.



Miguel Angel Garcia received the B.S., M.S., and Ph.D. degrees in computer science from Polytechnic University of Catalonia, Barcelona, Spain, in 1989, 1991, and 1996, respectively. He joined the Department of Software at Polytechnic University of Catalonia in 1996 as an Assistant Professor. From 1997 to 2006, he was with the Department of Computer Science and Mathematics at Rovira i Virgili University, Tarragona,

Spain, where he was the Head of Intelligent Robotics and Computer Vision Group. In 2006, he joined the Department of Informatics Engineering at Autonomous University of Madrid, Spain, where he is currently Associate Professor. His research interests include image processing, 3-D modeling, and mobile robotics.



Domenech Puig received the M.S. and Ph.D. degrees in computer science from Polytechnic University of Catalonia, Barcelona, Spain, in 1992 and 2004, respectively. In 1992, he joined the Department of Computer Science and Mathematics, Rovira i Virgili University, Tarragona, Spain, where he is currently Associate Professor. Since July 2006, he is the Head of the Intelligent Robotics and Computer Vision Group at the

same university. His research interests include image processing, texture analysis, perceptual models for image analysis, scene analysis, and mobile robotics.