Pattern Recognition 43 (2010) 3282–3297

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Application-independent feature selection for texture classification

Domenec Puig[a,*], Miguel Angel Garcia[b], Jaime Melendez[a]

[a] Intelligent Robotics and Computer Vision Group, Department of Computer Science and Mathematics, Rovira i Virgili University, Av. Paisos Catalans 26, 43007 Tarragona, Spain
[b] Department of Informatics Engineering, Autonomous University of Madrid, Francisco Tomas y Valiente 11, 28049 Madrid, Spain

ARTICLE INFO

Article history:
Received 5 September 2008
Received in revised form
28 February 2010
Accepted 5 May 2010

Keywords:
Texture feature selection
Supervised texture classification
Multiple texture methods
Multiple evaluation windows

ABSTRACT

Recent developments in texture classification have shown that the proper integration of texture methods from different families leads to significant improvements in terms of classification rate compared to the use of a single family of texture methods. In order to reduce the computational burden of that integration process, a selection stage is necessary. In general, a large number of feature selection techniques have been proposed. However, a specific texture feature selection must be typically applied given a particular set of texture patterns to be classified. This paper describes a new texture feature selection algorithm that is independent of specific classification problems/applications and thus must only be run once given a set of available texture methods. The proposed application-independent selection scheme has been evaluated and compared to previous proposals on both Brodatz compositions and complex real images.

1. Introduction

A texture classification problem consists of determining the textures present in an image given a set of texture patterns of interest. Many texture classification problems usually require the computation of a large amount of texture features in order to characterize their associated patterns. This implies that texture classifiers frequently combine big sets of features without taking into account their relevance and redundancy. Thus, lowering the dimensionality of a feature set is necessary for preserving the most relevant features and reducing the computational cost derived from unnecessary features that do not contribute to increasing the quality of the available information for each class [1–3].

Feature selection algorithms aim at choosing a subset of relevant features that are sufficient to capture the important properties of the set of classes that are to be distinguished. In the scope of classification, feature selection aims at selecting the smallest subset of features such that [1,4]: classification accuracy does not significantly decrease, and the resulting distribution of classes is as close as possible to the distribution obtained after considering all the available features.

Therefore, feature selection algorithms can be applied in order to obtain a significant subset of texture features given a specific texture classification problem. However, there is no consensus on which of the existing feature selection techniques is the most appropriate, since it is necessary to tune many low level parameters related to the design of the algorithms themselves in order that they produce acceptable results [5]. Currently, feature selection for texture classification is commonly tailored to the particular *texture feature extraction methods* (*texture methods*) that are available and, especially, to the particular texture patterns to which the different proposals are targeted.

This paper extends the selection scheme introduced in [6] and proposes a new application-independent strategy for the selection of texture feature extraction methods that determines a suitable subset of texture methods applicable to different texture classification problems. In this way, given any set of available texture methods, the proposed methodology automatically determines a reduced subset of them whose integration produces good classification results independently of the texture classification problem to be solved and, thus, of the texture patterns to be classified.

Experimental results with complex textured images belonging to different domains show that the proposed selection scheme leads to classification results comparable to those obtained when a specific selection of texture methods is performed for each particular classification problem. Furthermore, the selection methodology presented in this paper along with the classifier introduced in [7] produces better classification results than well-known texture classifiers based on texture methods belonging to the same family.

The remainder of this paper is organized as follows. A review of previous work on feature selection for classification is described in Section 2. Section 3 summarizes the texture classifier

[*] Corresponding author. Tel.: +34 977 55 96 77; fax: +34 977 55 97 10.
E-mail addresses: domenec.puig@urv.cat, dpuig@etse.urv.es (D. Puig), miguelangel.garcia@uam.es (M.A. Garcia), jaime.melendez@urv.cat (J. Melendez).

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.05.005

that is the basis for the proposed technique. Section 4 describes the proposed application-independent texture feature selection scheme. Section 5 shows experimental results of the selection and integration of widely used texture methods with the proposed technique, as well as a comparison with a well-known texture classification framework (*MeasTex* [8]) and the LBP classifiers [9]. Conclusions and further improvements are finally presented in Section 6.

2. Related work

Ideally, a feature selection method must evaluate all the possible subsets of features that can be generated from a given set of them and finally select the best. However, this procedure is exhaustive as it only tries to find the best subset. Hence, it may be too costly and prohibitive in practical terms. Other methods based on heuristic or random search try to reduce the computational complexity by compromising performance. These methods require a stopping criterion to prevent an exhaustive search of subsets. Therefore, a typical feature selection method has four basic steps [1]: *a search procedure* to generate the next candidate subset, an evaluation function to evaluate the subset under examination, a *stopping criterion* to decide when to stop, and a *validation procedure* to check whether the subset is valid or not.

Feature selection methods for classification can be categorized into two extensive groups according to their dependence on the algorithm that will finally use the selected subset: *filter methods*, which are independent of the classification algorithm, and *wrapper methods*, which use the classification algorithm as their evaluation function.

Filter methods have been more extensively developed than wrappers since the former ones are oriented to unsupervised problems, where no information about the classes is known a priori. However, if this information is available, a wrapper model can be more appropriate [10] than a filter one because the latter (e.g. [11]) totally ignores the effects of the selected feature subset on the performance of the classifier that will ultimately make use of those features.

More recent research on feature selection has focused on wrapper approaches (e.g. [12]) with the argument that, if the purpose of a selection task is to determine a feature subset useful for classification, the feature selection algorithm must take into account the biases of the classifier that will finally utilize the selected features for classifying unseen data. However, the most popular wrapper feature selection algorithms suffer from some of the general problems indicated in Section 1 and, in addition, they also have specific drawbacks. Thus, algorithms based on sequential forward or backward generation, such as WSFG and WSBG [1], are inefficient and unable to handle large training data sets in practice. Another problem is that they produce a ranked list of features, but the user must finally decide the number of features that are to be used.

Algorithms that follow a random (non-deterministic) search strategy, such as LVW [13], RMHC-PF1 [14] and others, require the proper assignment of values to different input parameters. Those parameters typically include the maximum number of iterations of the search process, a threshold for the evaluation function they use and a percentage in order to choose a portion of the training data proportional to that percentage for the first iteration of the selection algorithm. Furthermore, some algorithms require other specific parameters. For instance, LVW needs a threshold for the inconsistency rate. GA [1] needs an initial population size, a crossover rate and a mutation rate. SA [1] requires an annealing schedule, an initial temperature and a mutation probability.

The algorithms mentioned above, which perform a random generation of feature subsets, usually produce a minimum subset of features with no order among them. However, a feature selection algorithm that uses the same training data and keeps the same initial values for the input parameters can produce different minimum subsets of features at each execution. In this respect, many approaches select a feature subset that represents the average of the minimum subsets obtained after a certain number of independent executions of the selector (e.g. [1,15]).

Another problem of current feature selection algorithms is that they exhibit an increasingly poor performance when the number of relevant features in the initial feature set is large, as it can be appreciated in the experiments carried out in [15]. Furthermore, the ability to produce good feature subsets when the training data samples contain noise is another critical aspect that must be taken into account. For example, some algorithms, such as LVF, LVI and LVW [13], require the user to provide the noise level as a new input parameter.

Alternatively, the feature selection strategy presented in this paper follows the wrapper approach and overcomes some of the aforementioned drawbacks. Thus, the proposed scheme performs an automatic selection that is independent of the classification problem, does not require tuning initial parameters, is able to handle large training data sets, even in the presence of noise, and finally generates a deterministic minimum subset of sorted features.

3. Background

This section summarizes the pixel-based texture classifier introduced in [7], which is the basis for the feature selection scheme described in Section 4. This classifier is schematized in the flowchart shown in Fig. 1 (it will be referred to as CIMTF).

Let $\{\tau_1, \ldots, \tau_T\}$ be a set of T texture patterns of interest. Every texture τ_k is described by a set of sample images. Let $\mathbf{I}$ be an input textured image and let $\{\mu_1, \ldots, \mu_M\}$ be M texture methods that generate feature vectors when they are evaluated in the neighborhood of a certain pixel $\mathbf{I}(x,y)$ by using a set of W windows of different size, $\{s_1 \times s_1, \ldots, s_W \times s_W\}$. In order to classify a target pixel $\mathbf{I}(x,y)$, a set of $M \times W$ features is determined. Each feature $\mu_{ij}(\mathbf{I}(x,y))$ is obtained by applying a texture method μ_i to the pixels contained in a square window of size $s_j \times s_j$ centered at $\mathbf{I}(x,y)$.

The classification algorithm described in [7] consists of the following five stages:

(a) *Supervised training stage*: A set of $M \times W \times T$ likelihood functions $P_{ij}(\mathbf{I}(x,y)|\tau_k)$ are defined based on the probability distributions P_{ijk} corresponding to the evaluation of every

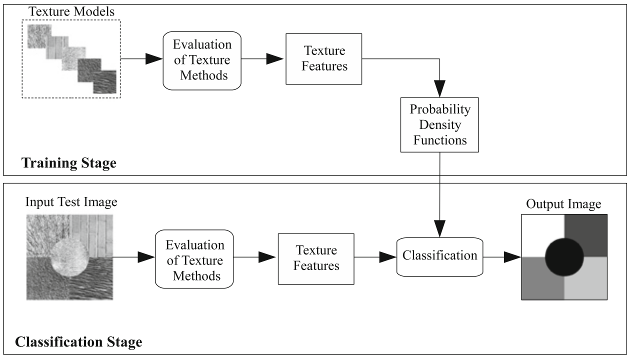

Fig. 1. Proposed scheme for supervised pixel-based texture classification by integration of multiple texture features (CIMTF).

method μ_{ij} over the pixels of the sample images corresponding to each texture τ_k. Each distribution is defined in the interval [MIN_{ijk},MAX_{ijk}]:

$$P_{ij}(\mathbf{I}(x,y)|\tau_k) = P_{ijk}(\mu_{ij}(\mathbf{I}(x,y)) \in [\mathrm{MIN}_{ijk},\mathrm{MAX}_{ijk}]) \quad (1)$$

(b) *Integration of multiple texture methods and window sizes*: A linear opinion pool combines the above likelihood functions:

$$P_i(\mathbf{I}(x,y)|\tau_k) = \sum_{j=1}^{W} w_{ijk} P_{ij}(\mathbf{I}(x,y)|\tau_k) \quad (2)$$

Every w_{ijk} is computed as the average of the *KJ*-divergence between τ_k and the other patterns:

$$w_{ijk} = d_{ijk} \Big/ \sum_{r=1}^{W} d_{irk}$$

$$d_{ijk} = \frac{1}{T-1} \sum_{l=1, l\neq k}^{T} KJ_{ij}(\tau_l,\tau_k) \quad (3)$$

The *Kullback J-divergence*, which measures the separability of two probability distributions is defined as $KJ_{ij}(\tau_a,\tau_b) = \int_0^1 (A-B)\log(A/B)\,du$, with A and B being defined from the previous probability distributions $A=P_{ija}(\mathrm{MAX}_{ija}u+\mathrm{MIN}_{ija}(1-u))$ and $B=P_{ijb}(\mathrm{MAX}_{ijb}u+\mathrm{MIN}_{ijb}(1-u))$.
Given the set of $M \times T$ likelihood functions $P_i(\mathbf{I}(x,y)|\tau_k)$ defined above, the likelihoods corresponding to the M texture methods associated with each texture patterns τ_k are integrated, leading to T combined likelihood functions, $P(\mathbf{I}(x,y)|\tau_k)$:

$$P(\mathbf{I}(x,y)|\tau_k) = \sum_{i=1}^{M} w_{ik} P_i(\mathbf{I}(x,y)|\tau_k)$$

$$w_{ik} = \sum_{j=1}^{W} w_{ijk} \Big/ \sum_{r=1}^{M}\sum_{j=1}^{W} w_{rjk} \quad (4)$$

(c) *Maximum a posteriori estimation*: A set of T posterior probabilities are computed by applying the Bayes rule:

$$P(\tau_k|\mathbf{I}(x,y)) = \frac{P(\mathbf{I}(x,y)|\tau_k)P(\tau_k)}{\sum_{l=1}^{T} P(\mathbf{I}(x,y)|\tau_l)P(\tau_l)}$$

$$P(\tau_k) = \sum_{i=1}^{M} w_{ik} \Big/ \sum_{i=1}^{M}\sum_{l=1}^{T} w_{il} \quad (5)$$

$\mathbf{I}(x,y)$ is likely to belong to the texture class τ_k with the maximum posterior probability $P(\tau_k|\mathbf{I}(x,y))$.

(d) *Significance test*: A significance level λ_k is defined based on two ratios utilized to characterize the performance of classifiers: *sensitivity* (S_n) and *specificity* (S_p). Pixel $\mathbf{I}(x,y)$ will be finally labeled as belonging to texture class τ_k iff $P(\tau_k|\mathbf{I}(x,y)) > \lambda_k$. Otherwise, it will be classified as unknown.

(e) *Conflict resolution*: The technique detailed in stage (b) may be modified by introducing an optional conflict resolution stage that leads to a more accurate definition of the weights used during the integration stage, as described in [16]. This stage yields a noticeable improvement of classification rates.

4. Application independent texture feature selection

This section extends the texture feature selection algorithm first introduced in [6], which is intended as a preprocessing stage of the CIMTF classifier described in the previous section in order to select a subset of the given texture methods whose integration yields classification results similar to those obtained when all methods are integrated. The main contribution of the present paper is that the feature selection process is not based on the texture patterns that are to be classified in a particular application, but on general texture patterns that are independent of specific applications. Hence, the selection process must be applied just once provided a set of texture methods to be integrated, while the technique described in [6] must be rerun whenever there is a change in the texture patterns to be classified in a specific application.

A key aspect of the selection scheme proposed in this paper lies on the conclusions derived from the psychophysical experiments carried out in [17], which addressed the important issue of deciding what high-level perceptual features are relevant in texture perception and how they are used by humans. Those experiments concluded the existence of a set of perceptual features, such as *contrast*, *repetitivity*, *granularity* and others, which capture different aspects of texture descriptions. In the same work, a set of texture patterns from the Brodatz album [18] were proposed as representatives of the aforementioned perceptual features that humans can perceive in natural textures (Fig. 2 shows examples of some of those patterns).

In fact, the work presented by Rao and Lohse in [17] is considered as a pioneering reference in the literature related to perceptual texture analysis, since they established the basis for a standard computational representation of texture. In this way, that study has been extensively referenced in the literature by recent works on texture description and representation (e.g. [19,20]), texture retrieval (e.g. [21]), texture synthesis (e.g. [22]), and color texture segmentation (e.g. [23]).

Feature selection for image classification based on perceptual criteria is an incipient research field that has started to receive considerable attention in the literature during the last recent years, mainly for the classification of whole images or individual pixels, or to search images in databases by content, although not necessarily using texture as the only visual cue (e.g. [24,25]).

In this work, we propose the application of a texture feature selector to the perceptual features proposed in [17], with the purpose of choosing the subset of texture methods that is most suitable, a priori, for different pixel-based texture classification problems. The final goal consists of reducing the number of texture methods integrated by the classifier during the classification of new input images even if there is a slight reduction of the final classification rate with respect to the maximum rate obtained when all the available features are integrated. In some cases, though, such a reduction may lead to an increase of the classification rate due to the curse of dimensionality problem.

4.1. Texture feature selection algorithm

This section details the proposed texture feature selection algorithm and its integration with CIMTF. The selection process is executed only once before the first stage of the classifier (supervised training stage), since the selector utilizes training data independent of the information of interest for each particular texture classification problem.

In order to select a subset of texture methods for each window size, the significance of every method is first determined based on its performance in classifying the aforementioned application-independent perception-based patterns. For each different window size, all texture methods are sorted in descending order of significance. Finally, a sequential forward generation procedure keeps adding new methods from the top of each sorted list until a performance criterion is maximized.

An important aspect in the proposed selection technique consists of the proper definition of the *individual significance* of every texture method with respect to the set of available methods

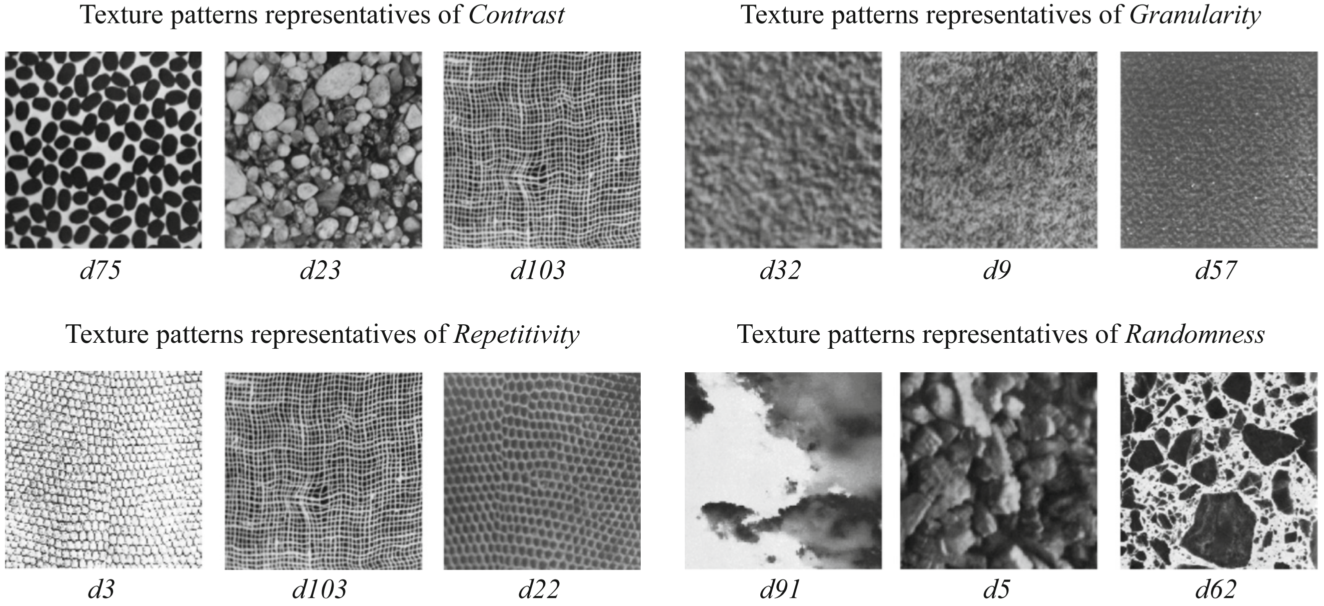
Texture patterns representatives of *Contrast*
d75
d23
d103
Texture patterns representatives of *Granularity*
d32
d9
d57
Texture patterns representatives of *Repetitivity*
d3
d103
d22
Texture patterns representatives of *Randomness*
d91
d5
d62

Fig. 2. Examples of application-independent texture patterns of interest from the human perception point of view. These textures were selected from the Brodatz album according to [17].

and the set of independent perception-based texture patterns. The significance of each individual texture method is measured in terms of classification rates. For this reason, it is assumed that suitable training samples for the perceptual texture patterns are available, and that the underlying class-conditional probabilities (1) can be obtained a priori.

Therefore, as proposed in [17], a set of L high-level perceptual features $\{F_1,\ldots,F_L\}$ (with $L=12$) are considered. Each high-level perceptual feature F_c, $1 \le c \le L$, is represented by a specific set of N application-independent texture models $\{T_c^1,\ldots,T_c^N\}$, with every texture model T_c^z being associated with a sample image $\mathbf{I}_c^z$, $1 \le z \le N$, according to [17].

Given any set of M texture methods $\{\mu_1,\ldots,\mu_M\}$, the proposed technique is defined as follows. Let $\mu_{ij}(\mathbf{I}_c^z(x,y))$ be the value of texture method μ_i, $1 \le i \le M$, when applied to a certain pixel $\mathbf{I}_c^z(x,y)$ of a sample image $\mathbf{I}_c^z$, which represents perceptual feature F_c, by using a window of size $s_j \times s_j$. This value will range in a real interval $[\mathrm{MIN}_{ijc},\mathrm{MAX}_{ijc}] : \mathrm{MIN}_{ijc} \le \mu_{ij}(\mathbf{I}_c^z(x,y)) \le \mathrm{MAX}_{ijc}, \quad \forall z \in [1,N]$. It is possible to determine a discrete probability function P_{ijc} associated with the values of μ_{ij} when measured over data samples that illustrate feature F_c, once those values are discretized into a natural range $[0,\beta)$:

$$P_{ijc}(s) = \sum_{z=1}^{N} \frac{|\mathbf{I}_c^z(x,y)|}{|\mathbf{I}_c^z|}$$

$$\forall \mathbf{I}_c^z(x,y) / h_{ijc}(\mu_{ij}(\mathbf{I}_c^z(x,y))) = s \qquad (6)$$

where $s \in [0,\beta)$, $\|$ is the cardinality of a set (number of pixels in case of an image) and $h_{ijc}(r)$ is defined as

$$h_{ijc}(r) = \left\lfloor (\beta-1)\frac{r-\mathrm{MIN}_{ijc}}{\mathrm{MAX}_{ijc}-\mathrm{MIN}_{ijc}} \right\rfloor r \in [\mathrm{MIN}_{ijc},\mathrm{MAX}_{ijc}] \qquad (7)$$

It is assumed that these probabilities are obtained by using the sufficient number of data to minimize the curse of dimensionality problem. The CIMTF classifier uses those probability distributions in order to classify each pixel of the $N \times L$ training image samples into one of the L known application-independent perceptual classes. Then, it is possible to compare the output of the classifier with the class label associated with each training sample and, therefore, calculate the classification rate R_{ijc}, $R_{ijc} \in [0,100]$, associated with the set of N texture classes T_c^z corresponding to perceptual feature F_c.

The *individual significance* S_{ijc} of a texture method μ_i with respect to the set of N texture patterns T_c^z associated with feature F_c and window size $s_j \times s_j$ is defined as the normalized classification rate:

$$S_{ijc} = R_{ijc} \Big/ \sum_{m=1}^{M} R_{mjc}, \quad S_{ijc} \in [0,1] \qquad (8)$$

S_{ijc} is defined in the interval [0,1], with zero indicating that perceptual feature F_c cannot be distinguished by using texture method μ_{ij}, and one that μ_{ij} is the only method through which F_c can be identified. Finally, the *global significance* S_{ij} of method μ_{ij} by considering all the aforementioned application-independent classes is formulated as

$$S_{ij} = \sum_{c=1}^{L} S_{ijc} \qquad (9)$$

Taking (9) into account, a ranking of all the M available texture methods is defined for every window size. The method at the top of each list is the one with the highest global significance S_{ij}, while the last method in each list is the one with the lowest S_{ij}. Thus, a list sorted in descending order is created for each window size.

The final goal consists of choosing a specific subset of texture methods out of each sorted list. In order to do this, every pixel in the $N \times L$ training images is subsequently classified for each specific window size. The mean classification rates are then computed by using different subsets of methods, starting with the most significant method and progressively adding a new method from the head of the sorted list until all methods are considered. Let Φ and ϕ, respectively, be the maximum and minimum mean classification rates obtained after the previous iterative process.

Let $R_{[1,m]jc}$ be the percentage of training samples associated with feature F_c that are correctly classified (classification rate) into the texture classes T_c^z when the first m methods, $m \in [1,M]$, from the head of the sorted list (m most significant methods) for windows of size $s_j \times s_j$ are utilized. Let $R_{[1,m]j}$ be the mean classification rate for the known application-independent classes corresponding to L high-level perceptual features obtained by integrating the m most significant methods.

Both $R_{[1,m]j}$ and m are normalized between zero and one: $\overline{R_{[1,m]j}}=(R_{[1,m]j}-\phi)/(\Phi-\phi)$ and $\overline{m}=(m-1)/(M-1)$. A performance measure ranging between zero and one is then defined as

$$\rho_{[1,m]j}=\frac{1}{3}\left(2\frac{\overline{R_{[1,m]j}}+1}{\overline{m}+1}-1\right) \tag{10}$$

which yields its maximum value ($\rho_{[1,m]j}=1$) when the maximum classification rate Φ is obtained with the first, most significant method ($\Phi=R_{1j}$), and its minimum value ($\rho_{[1,m]j}=0$) if the minimum rate ϕ is obtained when all the M available methods are integrated ($\phi=R_{Mj}$). In the end, the subset constituted by the η_j most significant methods for windows of size $s_j \times s_j$ is chosen, with η_j being

$$\eta_j=\underset{m}{\operatorname{argmax}} R_{[1,m]j}\rho_{[1,m]j} \tag{11}$$

In order to reduce the computational cost of an exhaustive search along the feature space, the proposed feature selection technique does not guarantee an optimal subset of texture methods, but a sub-optimal subset in the sense that it leads to classification rates similar to the ones obtained when all the available methods are used by CIMTF, as it is shown in the experiments. The performance criterion defined in (10) yields its maximum value when the classifier obtains a classification rate near the peak rate by using as few methods as possible. Therefore, this selection scheme leads to a trade-off solution between computational efficiency and classification quality.

5. Experimental validation

Different experiments have been carried out in order to validate the performance of the proposed application-independent selection scheme for supervised texture classification. The following sections introduce the framework utilized in this work and show experimental results. Finally, these results are analyzed and discussed.

5.1. Experimental setup

The proposed feature selection scheme has been applied in order to select, for different window sizes, respective subsets of texture methods from among an initial set of 68 texture methods. The aforementioned application-independent texture selector automatically determines a subset of the given texture methods that yields classification results for different problems similar to or even better than the results obtained by integrating the 68 methods. Every problem involves different application-dependent texture patterns to be recognized.

The proposed technique has been evaluated over a broad collection of test images corresponding to five different groups of images: well-known Brodatz [18] compositions based on the ones used in [26], outdoor images taken both at ground level and by aerial devices, VisTex [27], MeasTex [8], and the Berkeley database [28]. Those input test images have been used for evaluating the results of the CIMTF classifier, when it integrates the subset of texture methods selected by the technique described in Section 4.

As an example, this section reports the results obtained for the 25 test images shown in Figs. 3–7. This set of images used to validate the proposed methodology is representative of different texture classification problems, since a large amount of different texture patterns are present in every group of test images. The full set of experimental results produced by the proposed technique is available at http://deim.urv.cat/rivi/ifs.html.

Considering previous surveys on texture analysis and classification (e.g. [26]), 68 widely used texture methods have been considered to be integrated with CIMTF. The evaluated texture methods have been organized into three broad classes: "Classical" methods, optimized Gabor wavelet filters proposed in [29], which will be denoted as "GMM", and Local Binary Patterns, denoted as "LBP" [9].

While the last two classes constitute families of methods by themselves, the "Classical" class includes heterogeneous families of methods. In particular, 14 methods belonging to five different families are considered per window size [26]: four *Laws filter masks* (*R5R5*, *E5L5*, *E5E5*, *R5S5*), two *wavelet transforms* (*Daubechies-4, Haar*), four non-optimized *Gabor filters* with different wavelengths (8, 4) and orientations (0, 45, 90, 135), three *statistics* (*variance*, *skewness*, *homogeneity*) and the *fractal dimension.*

The GMM class is built as a filter bank with six scales and four orientations. The texture features that characterize every pixel and its surrounding neighborhood are the mean and standard

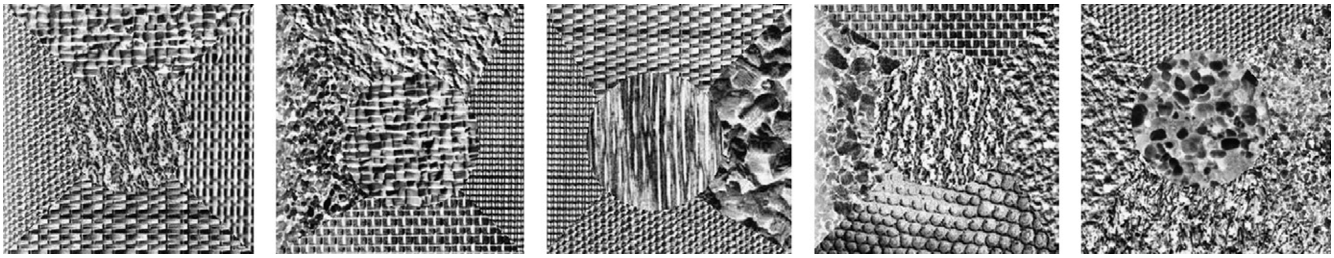

Fig. 3. Test images with portions of Brodatz texture patterns.

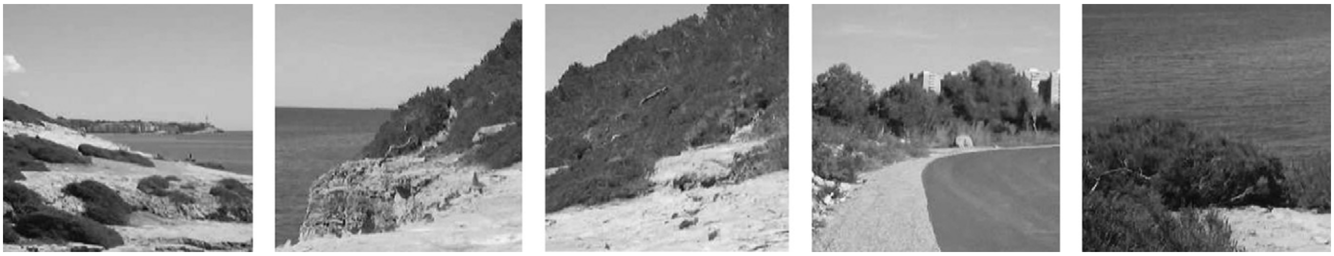

Fig. 4. Test images with outdoor ground scenes.

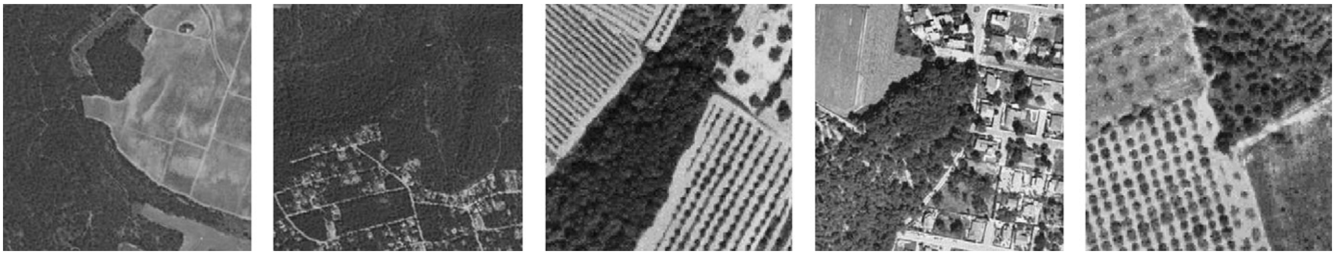

Fig. 5. Test images with aerial scenes.

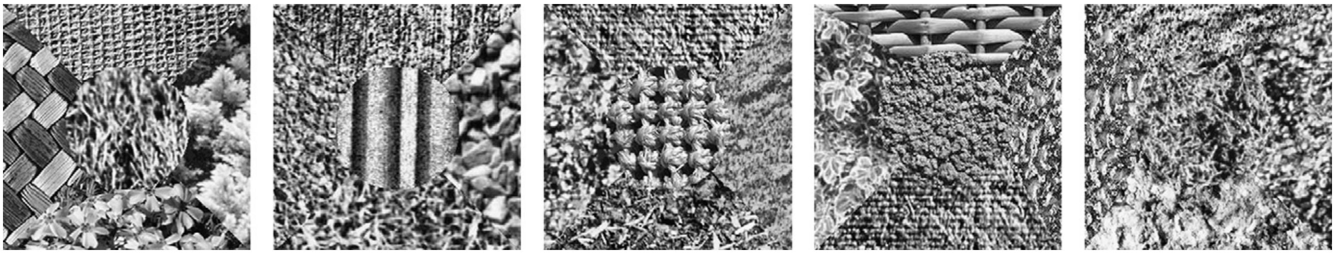

Fig. 6. Test images corresponding to the VisTex and MeasTex databases.

Fig. 7. Test images corresponding to the Berkeley database.

deviation of the module of the Gabor wavelet coefficients obtained after filtering the input image. Thus, the GMM class encompasses a total of $6 \times 4 \times 2 = 48$ methods. These settings come from [29,30]. The kernel size is set to be the same as the window size.

The LBP class is constituted by the three uniform rotation invariant Local Binary Pattern operators [9]. The texture features used in these experiments are the mean and standard deviation of the values produced after applying each LBP operator to an input image for every considered window size. Hence, the number of texture methods corresponding to the LBP class becomes six (3×2).

It is important to note that the proposed selection technique does not require that these are the methods that are always integrated. Indeed, it can be applied to any other given set of texture feature extraction methods. Actually, those 68 methods have been chosen just as an example. The selection of a subset of texture methods from among those 68 methods has been repeated six times, every time with a different window size. The following window sizes have been considered: $\{3\times 3, 5\times 5, 9\times 9, 17\times 17, 33\times 33, 65\times 65\}$.

Results with the proposed scheme have also been compared to the ones produced by the texture classifiers included in MeasTex [8], a well-known texture classification framework that provides a set of supervised texture classifiers based on the combination of a single family of texture methods (e.g., Gabor, Markov, Fractal, Grey-Level Co-occurrence Matrices) and a pattern classifier (e.g., Multivariate Gaussian Bayes, MVG, *K*-nearest neighbors, KNN). This framework is oriented to the classification of whole images instead of individual pixels. Thus, in order to achieve pixel-based classification, it was utilized to classify every pixel of the test images given a subimage of size 33×33 centered at that pixel (33×33 is the default window size used in MeasTex).

Furthermore, in order to better assess the performance of the proposed methodology, another comparison in terms of classification rates has also been performed with a straightforward extension to pixel-based classification of the LBP-based classifier proposed in [9], using its local binary pattern (LBP) operators as texture feature extractors and the *G* statistic as a dissimilarity measure. This extension has been performed similarly to the one described for MeasTex, but with subimages of 16×16 as suggested in [9], and keeping all texture feature vectors as texture models.

5.2. Results and discussion

Taking into account the aforementioned 68 texture methods, Table 1 shows the rankings of methods and the methods finally selected by the proposed selector for the different window sizes when the selector utilizes the application-independent texture patterns associated with 12 perceptual features [17]: *contrast*, *repetitivity*, *granularity*, *randomness*, *smoothness*, *density*, *directionality*, *complexity*, *coarseness*, *regularity*, *orientation* and

Table 1
Descending order rankings of texture methods obtained by the proposed feature selection algorithm for each of the six window sizes utilized for evaluating a set of 68 texture methods on application-independent texture patterns associated with 12 perceptual features. Only the first 30 methods in the top of the ranking are shown for each window size. Bold cells correspond to the selected methods according to the maximum weighted performances.

Window size					
3 × 3	**5 × 5**	**9 × 9**	**17 × 17**	**33 × 33**	**65 × 65**
Laws R5R5	**Laws R5R5**	**Laws R5R5**	**GMM$_{\text{Mean}}$3, 0°**	**GMM$_{\text{Std}}$5, 45°**	**GMM$_{\text{Std}}$5, 135°**
Laws R5S5	**GMM$_{\text{Std}}$5, 45°**	**GMM$_{\text{Mean}}$5, 135°**	**GMM$_{\text{Mean}}$5, 135°**	**GMM$_{\text{Std}}$5, 135°**	**Laws E5L5**
Gabor(8, 0°)	**Laws R5S5**	**GMM$_{\text{Std}}$5, 135°**	**GMM$_{\text{Mean}}$5, 90°**	**GMM$_{\text{Std}}$5, 90°**	**GMM$_{\text{Std}}$5, 45°**
GMM$_{\text{Mean}}$5, 135°	**GMM$_{\text{Mean}}$5, 135°**	**Laws R5S5**	**GMM$_{\text{Std}}$5, 135°**	**GMM$_{\text{Mean}}$5, 90°**	**GMM$_{\text{Std}}$5, 0°**
GMM$_{\text{Mean}}$5, 45°	**GMM$_{\text{Mean}}$5, 45°**	**GMM$_{\text{Std}}$5, 90°**	**GMM$_{\text{Std}}$5, 45°**	**Laws E5L5**	**GMM$_{\text{Mean}}$5, 0°**
LBPriu$_{\text{Std}}$162	**GMM$_{\text{Std}}$5, 135°**	**GMM$_{\text{Std}}$5, 0°**	**GMM$_{\text{Mean}}$5, 90°**	**GMM$_{\text{Std}}$5, 0°**	**GMM$_{\text{Mean}}$5, 90°**
LBPriu$_{\text{Std}}$243	**LBPriu$_{\text{Mean}}$243**	**GMM$_{\text{Mean}}$5, 0°**	**Laws E5L5**	**GMM$_{\text{Mean}}$3, 0°**	**Laws R5S5**
GMM$_{\text{Std}}$5, 135°	**GMM$_{\text{Mean}}$4, 45°**	**GMM$_{\text{Std}}$5, 45°**	**GMM$_{\text{Std}}$5, 90°**	**GMM$_{\text{Mean}}$5, 135°**	**GMM$_{\text{Mean}}$5, 135°**
LBPriu$_{\text{Mean}}$243	**GMM$_{\text{Mean}}$5, 0°**	**GMM$_{\text{Mean}}$5, 45°**	**GMM$_{\text{Std}}$5, 0°**	**Laws R5S5**	**GMM$_{\text{Std}}$5, 90°**
GMM$_{\text{Mean}}$5, 0°	**LBPriu$_{\text{Std}}$81**	**GMM$_{\text{Mean}}$5, 90°**	**GMM$_{\text{Mean}}$5, 45°**	**GMM$_{\text{Mean}}$5, 0°**	**LBPriu$_{\text{Mean}}$81**
LBPriu$_{\text{Mean}}$162	**GMM$_{\text{Mean}}$4, 135°**	**GMM$_{\text{Mean}}$4, 135°**	**Laws R5S5**	**GMM$_{\text{Mean}}$5, 45°**	**LBPriu$_{\text{Std}}$162**
GMM$_{\text{Std}}$5, 45°	**GMM$_{\text{Std}}$4, 0°**	**LBPriu$_{\text{Std}}$81**	**Laws R5R5**	**LBPriu$_{\text{Mean}}$162**	**GMM$_{\text{Mean}}$3, 0°**
GMM$_{\text{Std}}$5, 90°	**GMM$_{\text{Std}}$5, 0°**	**GMM$_{\text{Std}}$4, 0°**	**LBPriu$_{\text{Std}}$81**	**Wav. Daub. 4**	**GMM$_{\text{Mean}}$5, 45°**
GMM$_{\text{Mean}}$4, 0°	**GMM$_{\text{Mean}}$1, 135°**	**GMM$_{\text{Std}}$4, 90°**	**LBPriu$_{\text{Mean}}$162**	**GMM$_{\text{Mean}}$4, 135°**	**LBPriu$_{\text{Std}}$81**
GMM$_{\text{Mean}}$3, 135°	**GMM$_{\text{Mean}}$0, 45°**	**Wavelet Haar**	**Wavelet Haar**	**Laws R5R5**	**LBPriu$_{\text{Mean}}$162**
GMM$_{\text{Mean}}$3, 90°	**GMM$_{\text{Mean}}$0, 90°**	**GMM$_{\text{Mean}}$4, 45°**	**Wav. Daub. 4**	**LBPriu$_{\text{Std}}$81**	**Wav. Daub. 4**
GMM$_{\text{Mean}}$0, 90°	**GMM$_{\text{Mean}}$0, 135°**	**GMM$_{\text{Mean}}$4, 0°**	**GMM$_{\text{Mean}}$4, 0°**	**Wavelet Haar**	**LBPriu$_{\text{Mean}}$243**
GMM$_{\text{Mean}}$2, 90°	GMM$_{\text{Mean}}$1, 90°	**GMM$_{\text{Mean}}$3, 135°**	**GMM$_{\text{Mean}}$4, 90°**	**LBPriu$_{\text{Std}}$162**	**Laws E5E5**
GMM$_{\text{Mean}}$1, 0°	GMM$_{\text{Mean}}$0, 0°	**GMM$_{\text{Mean}}$4, 90°**	**GMM$_{\text{Mean}}$4, 135°**	**GMM$_{\text{Mean}}$4, 90°**	**GMM$_{\text{Mean}}$4, 135°**
GMM$_{\text{Mean}}$0, 45°	GMM$_{\text{Mean}}$2, 135°	**LBPriu$_{\text{Mean}}$162**	**LBPriu$_{\text{Mean}}$243**	**Variance**	**Laws R5R5**
GMM$_{\text{Mean}}$1, 135°	LBPriu$_{\text{Mean}}$162	**GMM$_{\text{Mean}}$0, 135°**	**GMM$_{\text{Std}}$4, 0°**	Laws E5E5	**Wavelet Haar**
GMM$_{\text{Mean}}$4, 135°	GMM$_{\text{Mean}}$1, 45°	GMM$_{\text{Mean}}$3, 0°	**Laws E5E5**	GMM$_{\text{Std}}$4, 90°	**GMM$_{\text{Mean}}$4, 90°**
GMM$_{\text{Mean}}$1, 90°	GMM$_{\text{Mean}}$4, 0°	GMM$_{\text{Mean}}$1, 135°	**LBPriu$_{\text{Std}}$162**	LBPriu$_{\text{Mean}}$243	**GMM$_{\text{Std}}$4, 90°**
GMM$_{\text{Mean}}$0, 135°	GMM$_{\text{Mean}}$1, 0°	GMM$_{\text{Std}}$0, 90°	**GMM$_{\text{Std}}$4, 90°**	GMM$_{\text{Mean}}$4, 45°	**GMM$_{\text{Mean}}$4, 45°**
GMM$_{\text{Mean}}$2, 135°	Wavelet Haar	GMM$_{\text{Mean}}$0, 0°	**GMM$_{\text{Mean}}$3, 90°**	GMM$_{\text{Mean}}$3, 90°	GMM$_{\text{Std}}$4, 45°
GMM$_{\text{Mean}}$1, 45°	GMM$_{\text{Mean}}$2, 45°	GMM$_{\text{Mean}}$0, 90°	**GMM$_{\text{Std}}$3, 0°**	GMM$_{\text{Std}}$4, 45°	Variance
GMM$_{\text{Mean}}$2, 0°	GMM$_{\text{Mean}}$2, 90°	GMM$_{\text{Mean}}$3, 45°	**GMM$_{\text{Mean}}$1, 45°**	Fractal	Fractal
GMM$_{\text{Mean}}$0, 0°	GMM$_{\text{Mean}}$2, 0°	GMM$_{\text{Mean}}$0, 45°	**GMM$_{\text{Std}}$0, 135°**	GMM$_{\text{Std}}$4, 0°	LBPriu$_{\text{Std}}$243
Gabor(4, 45°)	GMM$_{\text{Mean}}$3, 135°	LBPriu$_{\text{Mean}}$243	GMM$_{\text{Std}}$2, 0°	**GMM$_{\text{Mean}}$2, 0°**	GMM$_{\text{Mean}}$3, 90°
GMM$_{\text{Mean}}$3, 45°	GMM$_{\text{Std}}$4, 90°	GMM$_{\text{Mean}}$3, 90°	**GMM$_{\text{Mean}}$2, 0°**	GMM$_{\text{Std}}$4, 135°	GMM$_{\text{Std}}$2, 135°

uniformity. In particular, three images from the Brodatz album have been utilized to describe each of those perceptual features as suggested in [17]. Fig. 2 shows the images associated with some of those features.

The subset of texture methods selected in this way has been compared to the methods obtained when texture feature selection is performed for each different texture classification problem [6]. As an example, Tables 2 and 3 show the texture methods selected when the application-dependent texture patterns in Fig. 8 were, respectively, used during the selection process. Thus, Table 2 shows, for each window size, the top of the ranking of the 68 texture methods when the selection process is driven by the texture patterns associated with the first test image shown in Fig. 3. Table 3 presents similar information when the texture patterns used by the selector are those associated with the set of real outdoor test images shown in Fig. 4.

This example highlights that, when the selection process is driven by texture patterns corresponding to two different classification problems, the two sets of selected texture methods are considerably different (see Tables 2 and 3). In turn, the proposed application-independent selector chooses a set of methods (see Table 1) rather different to the two sets obtained for the two aforementioned problems.

However, some methods were selected in all cases, such as the non-optimized Gabor filters (Gabor 8, 0°) and the optimized Gabor wavelet filters (GMM$_{\text{Mean}}$5, 135°, GMM$_{\text{Mean}}$5, 45°, GMM$_{\text{Mean}}$5, 0°, GMM$_{\text{Std}}$5, 45°) for windows of 3 × 3 pixels, or Laws R5S5, GMM$_{\text{Std}}$5, 0°, GMM$_{\text{Std}}$5, 45°, GMM$_{\text{Mean}}$5, 0° for windows of 5 × 5 pixels, or Laws R5R5, GMM$_{\text{Mean}}$5, 45°, GMM$_{\text{Mean}}$5, 0° for windows of 9 × 9 pixels, among others.

This fact reinforces the evidence that although each specific problem ideally requires a particular subset of texture methods, there exist some methods that can be useful for a wide range of texture classification problems. In particular, the optimized Gabor wavelet filters with some specific scales and orientations have been selected the great majority of times.

On the other hand, Fig. 9 shows a comparison among the total number of texture methods selected in the three experiments discussed above (see Tables 1–3). As it can be appreciated, the number of selected methods associated with different application-dependent problems can be significantly different. Thus, a total of 55 texture methods were selected [6] for the classification of the first test image in Fig. 3, considering the six window sizes and the texture patterns shown in Fig. 8(top). As for the classification of the outdoor test images in Fig. 4, 179 methods were chosen by the selector in [6] with the selection process being driven by the texture models in Fig. 8(bottom).

In contrast, the methodology presented in this paper produces a stable result (128 texture methods were chosen among the set of 408 methods available in this work). Therefore, given an initial set of available texture methods, the proposed application-independent feature selector generates a unique subset of texture methods useful for the classification of a variety of test images involving different sets of texture models.

In this way, the experimental results presented in this section indicate that the proposed technique leads to a more general subset of texture methods that produce acceptable classification results when the CIMTF classifier, which integrates those selected

Table 2
Descending order rankings of texture methods obtained by the proposed feature selection algorithm for each of the six window sizes utilized for evaluating a set of 68 texture methods on the application-dependent sample images shown in Fig. 8(top). Only the first 20 methods in the top of the ranking are shown for each window size. Bold cells correspond to the selected methods according to the maximum weighted performances.

Window size					
3 × 3	**5 × 5**	**9 × 9**	**17 × 17**	**33 × 33**	**65 × 65**
Laws R5S5	**Laws R5S5**	**Laws R5S5**	**Laws R5S5**	**GMM$_{Mean}$5, 135°**	**GMM$_{Mean}$5, 135°**
Laws R5R5	**GMM$_{Mean}$5, 0°**	**GMM$_{Mean}$5, 45°**	**GMM$_{Mean}$5, 45°**	**Laws R5R5**	**Laws R5S5**
Laws E5L5	**Laws R5R5**	**GMM$_{Std}$5, 0°**	**GMM$_{Mean}$5, 135°**	**Laws R5S5**	**GMM$_{Mean}$5, 45°**
GMM$_{Mean}$5, 90°	**GMM$_{Std}$5, 0°**	**GMM$_{Mean}$5, 0°**	**Laws R5R5**	**GMM$_{Mean}$5, 45°**	**Laws R5R5**
LBPriu$_{Mean}$81	**GMM$_{Std}$5, 45°**	**GMM$_{Mean}$5, 135°**	**GMM$_{Mean}$5, 0°**	**GMM$_{Std}$5, 0°**	**GMM$_{Std}$5, 0°**
Laws E5E5	**GMM$_{Mean}$3, 90°**	**Laws R5R5**	GMM$_{Std}$5, 0°	**GMM$_{Mean}$5, 0°**	**Wavelet Haar**
Wavelet Haar	**GMM$_{Std}$3, 90°**	**GMM$_{Mean}$2, 90°**	GMM$_{Mean}$4, 0°	**Fractal**	**GMM$_{Std}$4, 45°**
GMM$_{Mean}$5, 45°	**Laws E5L5**	GMM$_{Std}$5, 45°	GMM$_{Mean}$3, 90°	LBPriu$_{Std}$81	**GMM$_{Mean}$5, 0°**
GMM$_{Std}$4, 90°	GMM$_{Mean}$5, 45°	GMM$_{Std}$5, 135°	GMM$_{Std}$5, 45°	LBPriu$_{Std}$162	**LBPriu$_{Std}$162**
GMM$_{Std}$5, 0°	GMM$_{Std}$5, 90°	GMM$_{Std}$3, 90°	LBPriu$_{Std}$81	GMM$_{Mean}$3, 90°	**Laws E5L5**
LBPriu$_{Std}$81	Laws E5E5	GMM$_{Std}$4, 0°	GMM$_{Std}$5, 135°	Wavelet Haar	**GMM$_{Mean}$2, 0°**
GMM$_{Std}$4, 135°	Wavelet Haar	GMM$_{Std}$3, 0°	GMM$_{Mean}$2, 90°	Laws E5L5	LBPriu$_{Std}$81
GMM$_{Std}$5, 45°	GMM$_{Mean}$4, 135°	GMM$_{Std}$0,90°	Laws E5E5	GMM$_{Std}$5, 45°	GMM$_{Mean}$1, 135°
GMM$_{Mean}$5, 135°	GMM$_{Mean}$4, 45°	GMM$_{Mean}$3, 90°	Laws E5L5	GMM$_{Std}$4, 45°	GMM$_{Mean}$3, 90°
Gabor (4, 90°)	GMM$_{Mean}$5, 90°	GMM$_{Mean}$2, 0°	GMM$_{Std}$0, 45°	GMM$_{Std}$5, 135°	GMM$_{Std}$2, 0°
Gabor (8, 0°)	GMM$_{Mean}$5, 135°	LBPriu$_{Std}$81	GMM$_{Std}$4, 0°	GMM$_{Mean}$4, 0°	GMM$_{Std}$5, 135°
GMM$_{Mean}$5, 0°	GMM$_{Std}$3, 0°	GMM$_{Std}$0, 45°	GMM$_{Std}$0, 135°	GMM$_{Std}$0, 45°	GMM$_{Mean}$3, 45°
Gabor (4, 45°)	GMM$_{Std}$5, 135°	GMM$_{Mean}$2, 45°	GMM$_{Mean}$2, 0°	GMM$_{Std}$1, 135°	GMM$_{Mean}$1, 90°
Gabor (8, 135°)	GMM$_{Std}$4, 0°	GMM$_{Std}$0, 135°	LBPriu$_{Std}$162	GMM$_{Mean}$1, 135°	GMM$_{Std}$3, 0°
GMM$_{Std}$5, 90°	LBPriu$_{Std}$81	Laws E5L5	GMM$_{Mean}$2, 135°	GMM$_{Mean}$2, 0°	GMM$_{Std}$5, 45°

Table 3
Descending order rankings of texture methods obtained by the proposed feature selection algorithm for each of the six window sizes utilized for evaluating a set of 68 texture methods on the application-dependent sample images shown in Fig. 8(bottom). Only the first 20 methods in the top of the ranking are shown for each window size. Bold cells correspond to some of the selected methods according to the maximum weighted performances. Note that, in this case, not all the selected methods are shown due to their large number.

Window size					
3 × 3	**5 × 5**	**9 × 9**	**17 × 17**	**33 × 33**	**65 × 65**
GMM$_{Mean}$5, 135°	**GMM$_{Mean}$5, 45°**	**GMM$_{Mean}$4, 45°**	**GMM$_{Mean}$2, 0°**	**GMM$_{Mean}$0, 90°**	**Gabor (4, 90°)**
GMM$_{Mean}$5, 45°	**GMM$_{Mean}$5, 135°**	**GMM$_{Mean}$4, 135°**	**GMM$_{Mean}$3, 90°**	**GMM$_{Mean}$2, 90°**	Gabor (4, 45°)
GMM$_{Mean}$5, 0°	**GMM$_{Mean}$5, 90°**	**GMM$_{Mean}$3, 45°**	**GMM$_{Mean}$1, 90°**	**GMM$_{Mean}$0, 0°**	Gabor (8, 0°)
GMM$_{Mean}$5, 90°	**GMM$_{Mean}$4, 0°**	**GMM$_{Mean}$3, 90°**	**GMM$_{Mean}$1, 45°**	Gabor(8, 135°)	Gabor(8, 135°)
GMM$_{Mean}$1, 0°	**GMM$_{Mean}$5, 45°**	**GMM$_{Mean}$0, 90°**	**GMM$_{Mean}$0, 135°**	Gabor(4, 90°)	**GMM$_{Mean}$3, 90°**
GMM$_{Mean}$1, 135°	**GMM$_{Mean}$4, 90°**	**GMM$_{Mean}$0, 135°**	**GMM$_{Mean}$1, 0°**	Gabor(4, 45°)	**GMM$_{Mean}$0, 90°**
GMM$_{Mean}$2, 135°	**GMM$_{Mean}$0, 135°**	**GMM$_{Mean}$3, 0°**	**GMM$_{Mean}$1, 135°**	Gabor(8, 0°)	**GMM$_{Mean}$5, 90°**
GMM$_{Mean}$2, 0°	**GMM$_{Mean}$0, 90°**	**GMM$_{Mean}$0, 45°**	Gabor(4, 90°)	**GMM$_{Mean}$0, 45°**	**GMM$_{Mean}$3, 135°**
GMM$_{Mean}$0, 45°	**GMM$_{Mean}$0, 0°**	**GMM$_{Mean}$0, 0°**	Gabor(4, 45°)	**GMM$_{Mean}$1, 90°**	Wavelet Haar
GMM$_{Mean}$0, 135°	**GMM$_{Mean}$0, 45°**	**GMM$_{Mean}$1, 135°**	Gabor(8, 135°)	**GMM$_{Mean}$0, 135°**	Wav. Daub. 4
GMM$_{Mean}$0, 90°	**GMM$_{Mean}$1, 135°**	**GMM$_{Mean}$3, 135°**	Gabor(8, 0°)	Wavelet Haar	Laws E5E5
GMM$_{Mean}$0, 0°	**GMM$_{Mean}$1, 90°**	**GMM$_{Mean}$1, 90°**	**GMM$_{Mean}$2, 90°**	Wav. Daub. 4	**GMM$_{Mean}$4, 135°**
GMM$_{Mean}$3, 135°	**GMM$_{Mean}$1, 0°**	**GMM$_{Mean}$4, 0°**	**GMM$_{Mean}$3, 0°**	**GMM$_{Mean}$2, 0°**	**GMM$_{Mean}$0, 0°**
GMM$_{Mean}$1, 90°	**GMM$_{Mean}$1, 45°**	**GMM$_{Mean}$1, 45°**	**GMM$_{Mean}$0, 90°**	**GMM$_{Mean}$1, 0°**	**GMM$_{Std}$3, 90°**
GMM$_{Mean}$1, 45°	**GMM$_{Mean}$4, 135°**	**GMM$_{Mean}$1, 0°**	**GMM$_{Mean}$0, 45°**	**GMM$_{Mean}$4, 135°**	**GMM$_{Std}$4, 135°**
GMM$_{Mean}$2, 90°	**GMM$_{Mean}$2, 90°**	**GMM$_{Mean}$4, 90°**	**GMM$_{Mean}$0, 0°**	**GMM$_{Mean}$4, 90°**	**GMM$_{Mean}$4, 90°**
GMM$_{Mean}$3, 0°	**GMM$_{Mean}$2, 135°**	**GMM$_{Mean}$5, 0°**	Wavelet Haar	**GMM$_{Std}$4, 90°**	**GMM$_{Std}$0, 45°**
GMM$_{Mean}$2, 45°	**GMM$_{Mean}$2, 0°**	**GMM$_{Mean}$5, 90°**	Wav. Daub. 4	Laws E5E5	**GMM$_{Mean}$2, 135°**
GMM$_{Mean}$3, 45°	**GMM$_{Mean}$2, 45°**	Gabor(4, 45°)	**GMM$_{Mean}$4, 90°**	**GMM$_{Mean}$5, 90°**	**GMM$_{Std}$4, 90°**
GMM$_{Mean}$3, 90°	**GMM$_{Mean}$3, 90°**	Gabor(4, 90°)	**GMM$_{Mean}$4, 135°**	**GMM$_{Std}$4, 135°**	**GMM$_{Mean}$4, 45°**

methods, is applied to different classification problems. Therefore, it is possible to avoid a specific selection of texture methods for each particular classification problem.

Another important advantage of the proposed application-independent selection is that it avoids the computational cost associated with the execution of a specific selection of texture methods for every different classification task. The computational complexity of each selection process is closely related to the cost of the CIMTF texture classifier (see [7] for the details regarding the computational complexity of CIMTF, which is the basis of the proposed selection methodology). In particular, the cost of the selection process up to the stage in which the *global significance* of every texture method is obtained is $O(N'T^2WM)$, where M is the number of available texture methods, W the number of different window sizes, T the number of texture patterns to be recognized and N' the number of pixels of the training images associated with every pattern. Furthermore, an additional cost for ranking the texture methods according to their significance and then selecting

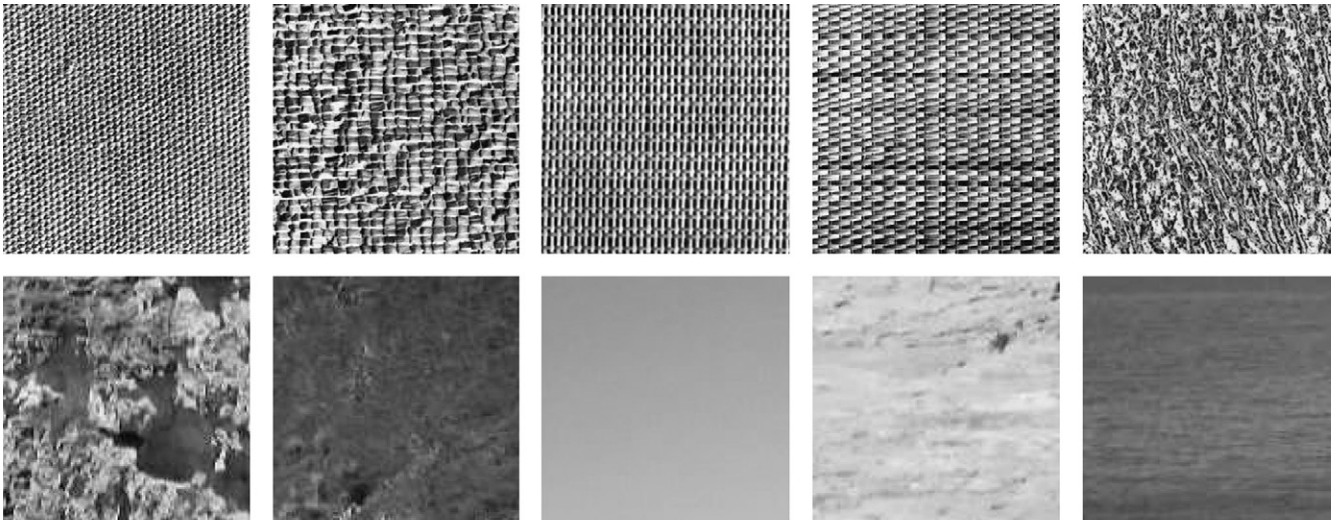

Fig. 8. Detail of application-dependent texture models: (top) pattern images from the Brodatz album associated with the first test image shown in Fig. 3; (bottom) outdoor patterns corresponding to the real scenes in Fig. 4.

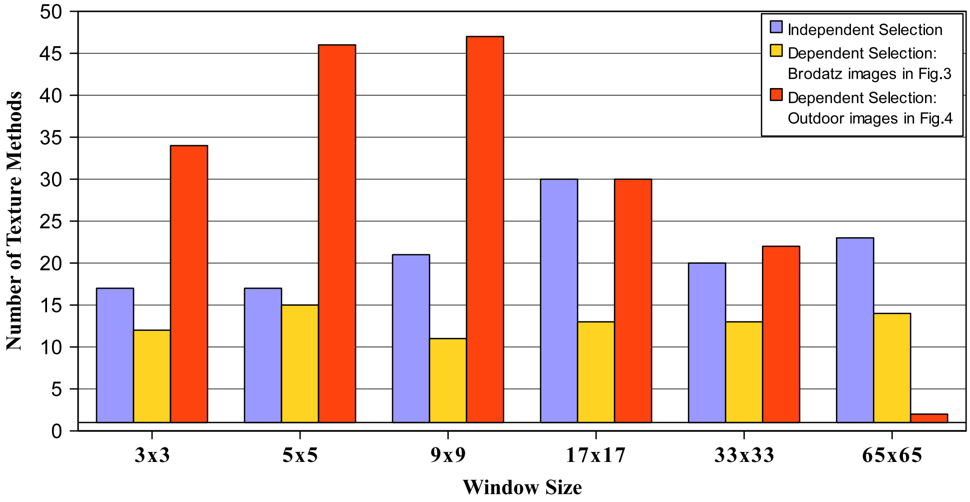

Fig. 9. Number of texture methods selected in different experiments for each window size.

the appropriate subset of methods at the top of the ranking must be considered.

The experiments performed in this paper have been run on a Pentium IV computer at 3.2 GHz. The time required for running each selection process associated with the experiments conducted in this work ranges from 117 to 289 min depending on the number of texture patterns and pixels per image. This clearly indicates the convenience of avoiding the execution of a particular selection of texture methods for every different set of texture patterns.

Figs. 10–14 depict the results obtained when the CIMTF classifier complemented with the new feature selection scheme presented in this paper is applied to the input images in Figs. 3–7. The first row in Figs. 10–14 shows the ground-truth classifications. The second row in Figs. 10–14 shows the classification maps when the classifier integrates the 68 available texture methods evaluated over the six different window sizes. The third row in Figs. 10–14 depicts the results when the classifier integrates, for every window size, the methods chosen by the proposed selector using application-independent texture patterns (see Table 1). The fourth row in Figs. 10–14 displays the results when the classifier integrates, for every window size, the texture methods selected by the feature selector described in [6] using application-dependent texture patterns. Finally, the fifth and sixth rows in Figs. 10–14 show the best results obtained with MeasTex and the LBP-based classifier proposed in [9], respectively.

The results in Figs. 10–14 indicate that the classifier fed with the texture methods chosen according to the selection scheme introduced in this paper yields classification rates similar or just slightly inferior to those achieved when all the available texture methods are integrated, but at a much lower cost (less than a third of methods), as shown in Table 4.

More importantly, it is possible to find a general subset of texture methods for each window size (see Table 1), with those methods being chosen with independence of the classification problem, which produces similar results (see row 3 in Figs. 10–14) to the ones obtained when a specific selection is carried out for each different classification problem by using application-dependent texture patterns (see row 4 in Figs. 10–14).

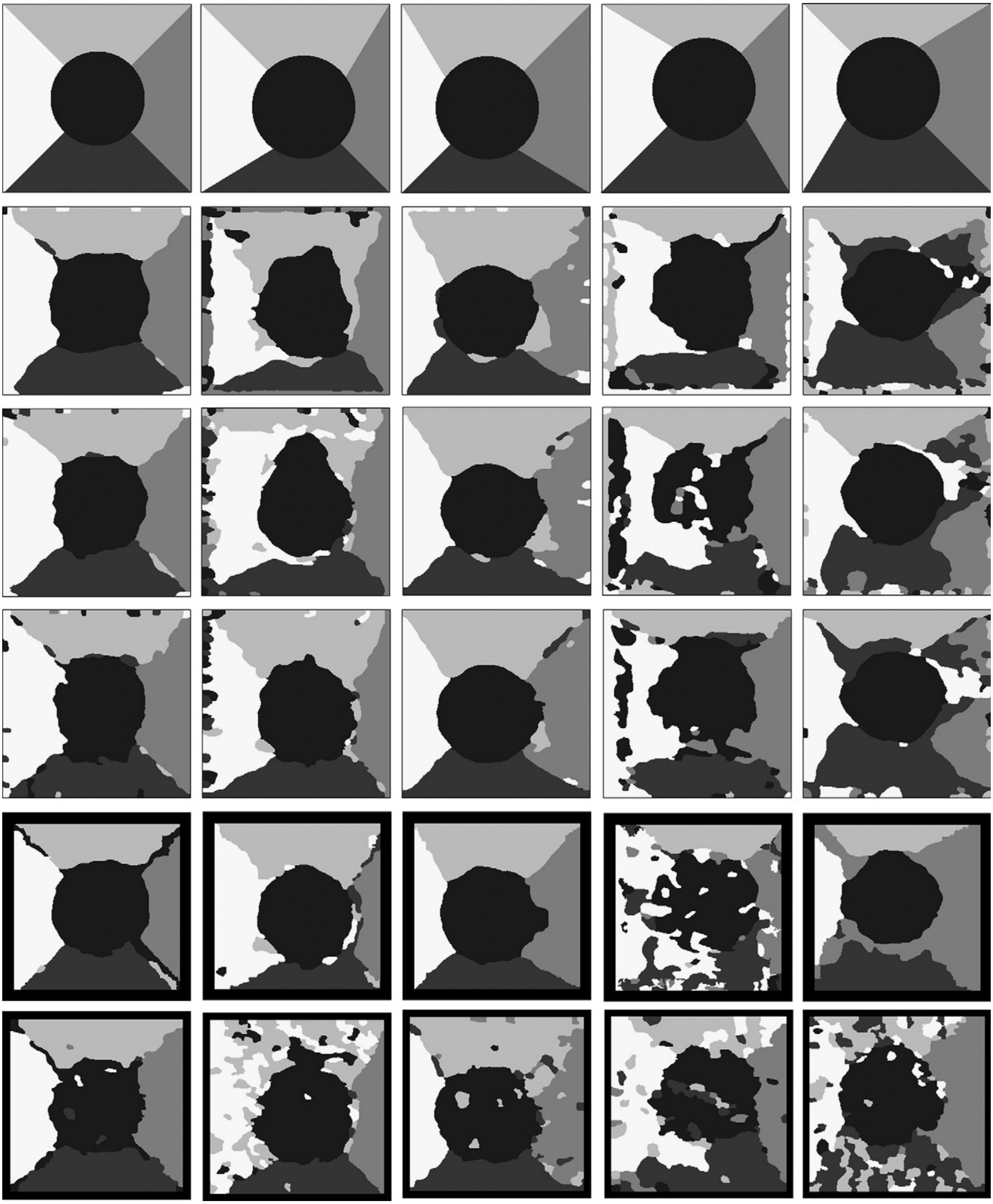

Fig. 10. (row 1) Ground-truth corresponding to test images in Fig. 3; classification maps with the CIMTF classifier, which integrates, for every window size: (row 2) the 68 given texture methods, (row 3) the texture methods selected by the proposed application-independent scheme, and (row 4) the texture methods specifically selected for each test image according to the selector in [6]; (row 5) best results with MeasTex in terms of classification rate; (row 6) best results with LBP in terms of classification rate.

Furthermore, the number of texture methods selected when a particular selection process is carried out for each classification problem is much more variable and unstable (see Table 4).

Finally, Table 5 shows that CIMTF fed with the texture methods selected by the proposed application-independent selector produces results slightly superior in terms of average

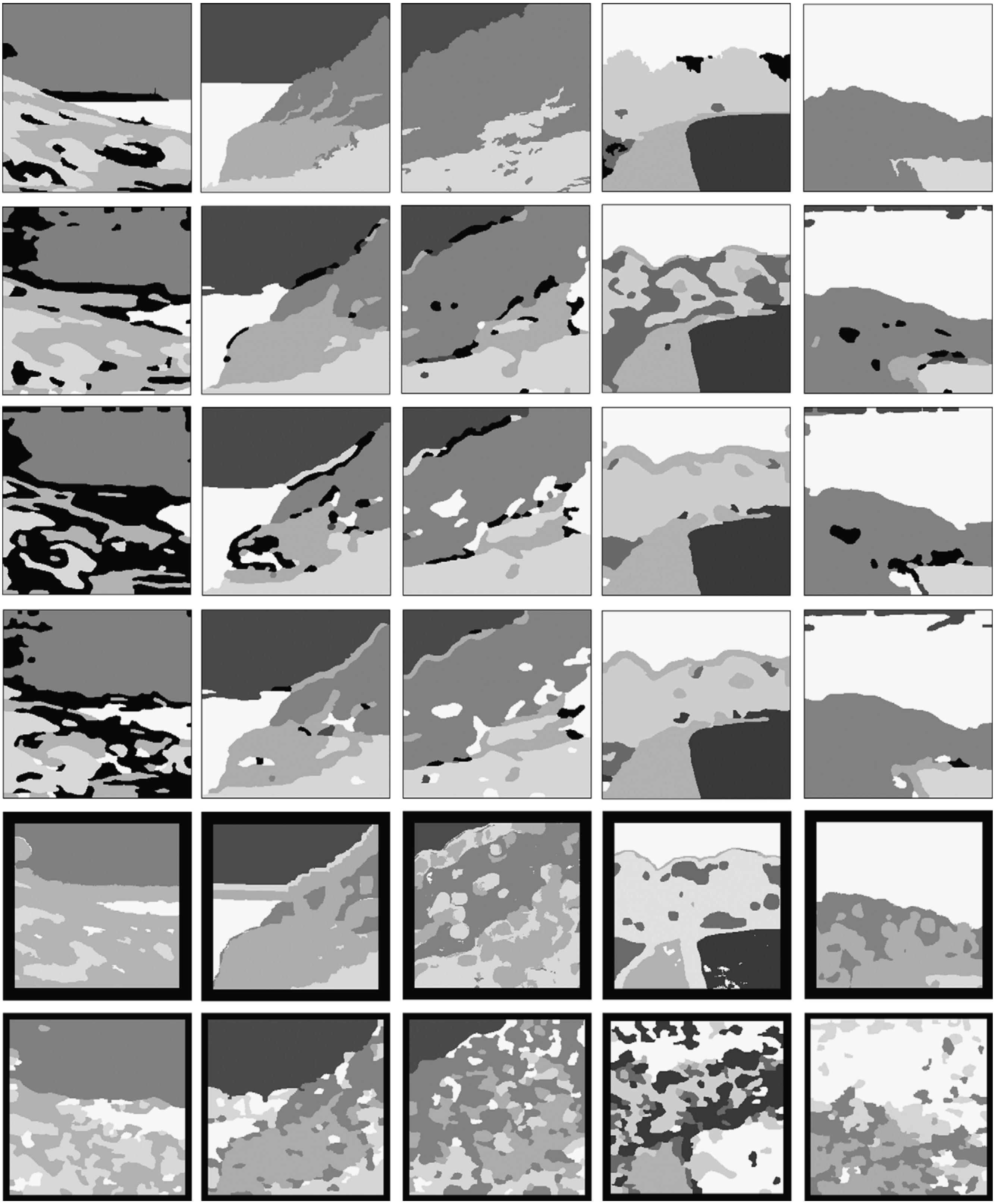

Fig. 11. (row 1) Ground-truth corresponding to test images in Fig. 4; classification maps with the CIMTF classifier, which integrates, for every window size: (row 2) the 68 given texture methods, (row 3) the texture methods selected by the proposed application-independent scheme, and (row 4) the texture methods specifically selected for each test image according to the selector in [6]; (row 5) best results with MeasTex in terms of classification rate; (row 6) best results with LBP in terms of classification rate.

classification rate to those obtained by the MeasTex and LBP classifiers. However, the best MeasTex result for each test image (shown in Figs. 10–14) was achieved with a different combination of texture method and classifier. For example, (Gabor, MVG) for the first and second test images in Fig. 3, (Markov, MVG) for the third and fifth, and (Gabor, KNN) for the fourth. The other

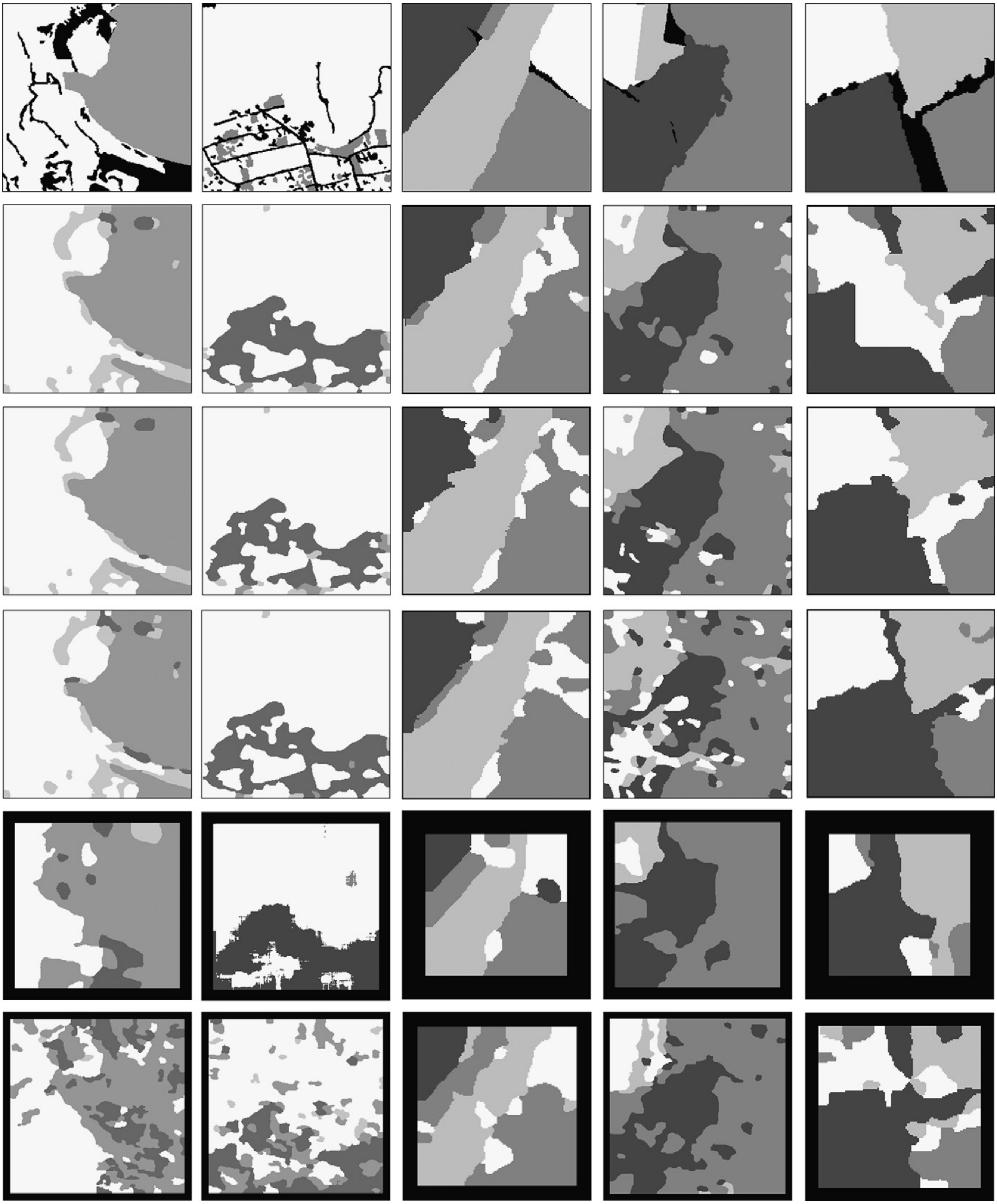

Fig. 12. (row 1) Ground-truth corresponding to test images in Fig. 5; classification maps with the CIMTF classifier, which integrates, for every window size: (row 2) the 68 given texture methods, (row 3) the texture methods selected by the proposed application-independent scheme, and (row 4) the texture methods specifically selected for each test image according to the selector in [6]; (row 5) best results with MeasTex in terms of classification rate; (row 6) best results with LBP in terms of classification rate.

combinations of texture families {Gabor, Fractal, Markov, GLCM} and classification algorithms {MVG, KNN} included in MeasTex produced much worse classification rates. Similar conclusions can be drawn from the LBP-based classifier, since the best classification rates were obtained with different Local Binary Pattern operators for each test image.

Notice that the MeasTex and LBP classifiers were unable to classify a strip of 16 and 8 pixels, respectively, which belongs to

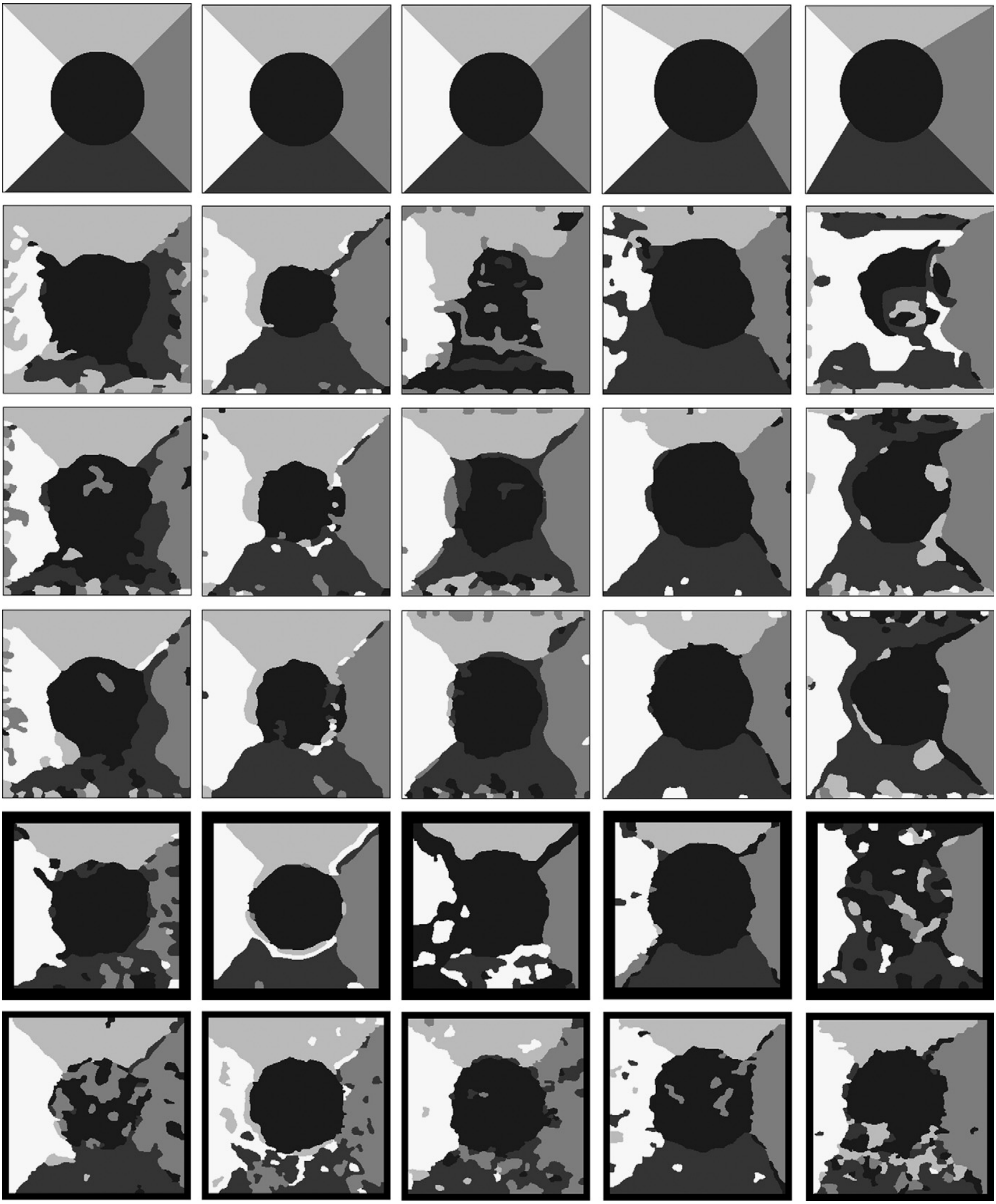

Fig. 13. (row 1) Ground-truth corresponding to test images in Fig. 6; classification maps with the CIMTF classifier, which integrates, for every window size: (row 2) the 68 given texture methods, (row 3) the texture methods selected by the proposed application-independent scheme, and (row 4) the texture methods specifically selected for each test image according to the selector in [6]; (row 5) best results with MeasTex in terms of classification rate; (row 6) best results with LBP in terms of classification rate.

the boundary of the images (see rows 5 and 6 in Figs. 10–14). The reason is that both MeasTex and LBP do not combine texture methods evaluated over windows of different size. Therefore, the classification rates corresponding to MeasTex and LBP have been evaluated without taking those strips into account (if the pixels in those strips were considered, the classification rates for MeasTex and LBP shown in Table 5 would be considerably lower). Furthermore, due to the use of windows of a single size, the

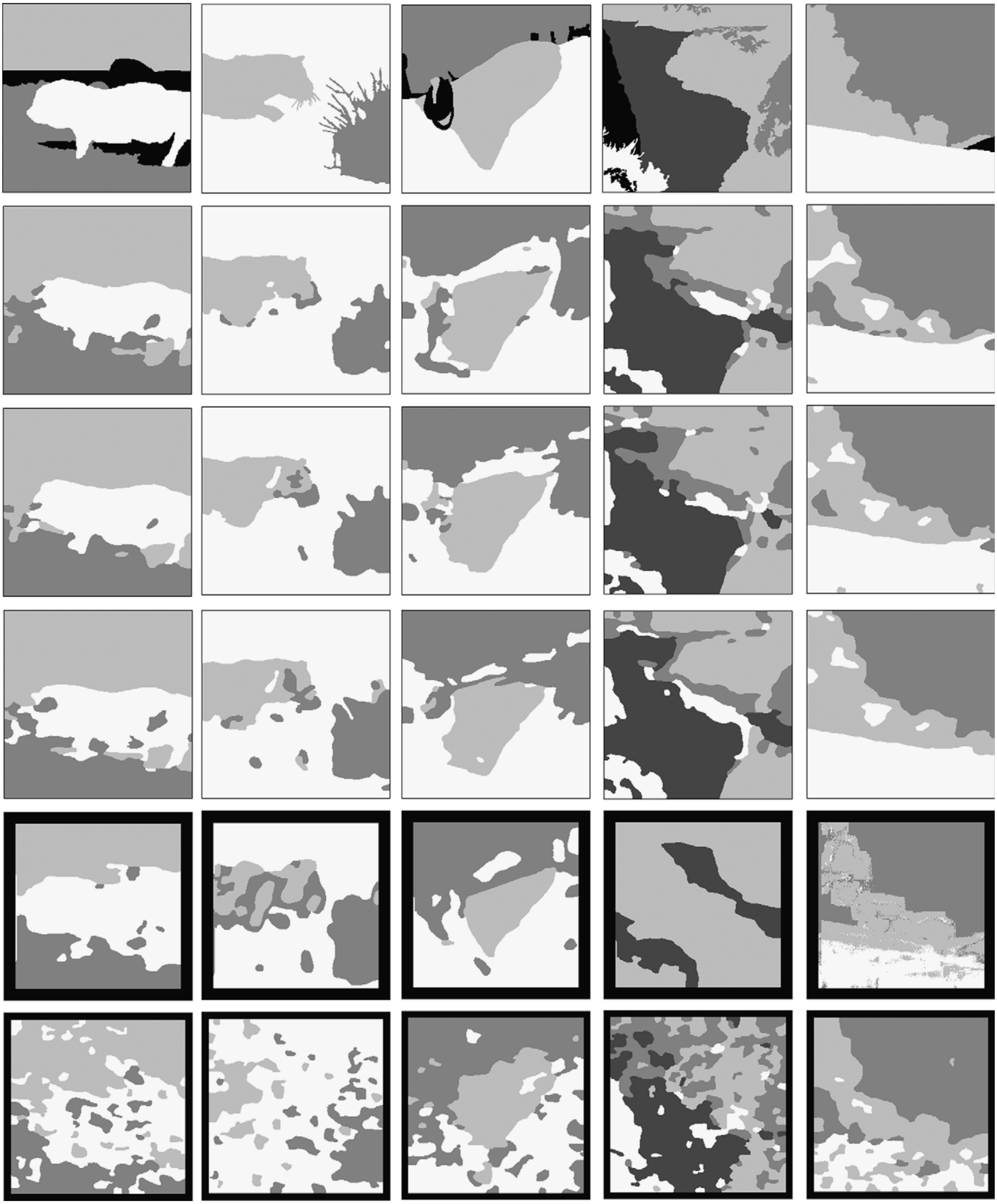

Fig. 14. (row 1) Ground-truth corresponding to test images in Fig. 7; classification maps with the CIMTF classifier, which integrates, for every window size: (row 2) the 68 given texture methods, (row 3) the texture methods selected by the proposed application-independent scheme, and (row 4) the texture methods specifically selected for each test image according to the selector in [6]; (row 5) best results with MeasTex in terms of classification rate; (row 6) best results with LBP in terms of classification rate.

Table 4
Classification rates (%) and number of texture methods integrated by the CIMTF classifier for the input images shown in Figs. 3–7. The first column corresponds to the classifier fed with all the texture methods and windows. The second column corresponds to the classifier fed with the texture methods selected by the proposed application-independent selector. The third column corresponds to the application of the feature selector in [6], which utilizes specific texture patterns for each classification problem. The number of selected methods is indicated between brackets.

Test image	CIMTF classifier when it integrates different subsets of texture methods for every window size		
	All methods and windows	Proposed application-independent methods (Table 1)	Application-dependent methods [6]
Average Fig. 10	81.92 (408)	82.25 (128)	86.06 (77)
Average Fig. 11	82.25 (408)	81.36 (128)	75.42 (179)
Average Fig. 12	85.03 (408)	88.00 (128)	85.32 (141)
Average Fig. 13	71.50 (408)	80.77 (128)	83.14 (83)
Average Fig. 14	84.89 (408)	85.53 (128)	85.05 (40)
Total average	**81.92 (408)**	**83.58 (128)**	**84.99 (104)**

Table 5
Classification rates (%) produced by the proposed scheme in comparison with those obtained by MeasTex and LBP for the input images shown in Figs. 3–7. The first column corresponds to the CIMTF classifier fed with the texture methods selected by using the proposed application-independent selector. The second column corresponds to the best results with MeasTex. The third column corresponds to the best results with LBP.

Test image	Classification technique		
	CIMTF (Section 3) with application-independent methods (Table 1)	Best classification rates with MeasTex	Best classification rates with LBP
Average Fig. 10	82.25	86.33	82.89
Average Fig. 11	81.36	69.56	62.75
Average Fig. 12	88.00	76.26	74.17
Average Fig. 13	80.77	77.97	83.95
Average Fig. 14	85.53	72.89	74.49
Total average	**83.58**	**76.60**	**75.65**

classification accuracy of MeasTex and LBP decreases in pixels close to boundaries among different regions.

6. Conclusions and further work

This paper describes a new feature selection methodology for texture image classification. For every window size, the proposed scheme automatically determines a reduced number of texture features whose integration produces classification results comparable to or better than those obtained when all the available features are utilized, but at a much lower computational cost. The main contribution of this work is the utilization as training data for the proposed feature selection algorithm of a set of application-independent texture patterns that represent the different features that humans can perceive in natural textures.

The advantage of this approach is that the feature selection process must be carried out only once given a set of texture feature extraction methods, with disregard of any particular classification problem, giving thus rise to a subset of texture methods that produce acceptable results for different classification problems, each involving different application-dependent texture patterns to be recognized. In this way, it is possible to avoid a specific selection of texture methods for every classification problem.

Experimental results with complex real outdoor images, and Brodatz, VisTex, MeasTex and Berkeley images show that a texture classifier which properly integrates texture methods selected for every window size by the proposed application-independent scheme outperforms widely recognized texture classifiers based on texture methods from a same family both quantitatively and qualitatively.

Further work will consist of extending the proposed technique to unsupervised segmentation. In addition, we also plan to extend this technique in order to be able to include in the selection process feature extraction methods originally devised for “macro-texture” analysis, such as the one presented in [31]. In this way, the proposed methodology would be able to select among a given set of both “micro-texture” and “macro-texture” feature extraction methods.

Acknowledgement

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

References

[1] M. Dash, H. Liu, Feature Selection for Classification. Intelligent Data Analysis, Elsevier, 1997, pp. 131–156.

[2] D. Koller, M. Sahami, Toward optimal feature selection, in: Proceedings of the 13th International Conference on Machine Learning, Bari, Italy, 1996, pp. 284–292.

[3] P. Zhang, J. Peng, B. Buckles, Learning optimal filter representation for texture classification, in: International Conference on Pattern Recognition, vol. 2, Hong Kong, 2006, pp. 1138–1141.

[4] X. Chen, X. Zeng, D. van Alphen, Multi-class feature selection for texture classification, Pattern Recognition Letters 27 (14) (2006) 1685–1691.

[5] H. Liu, L. Yu, Feature selection for data mining, Research Technical Report, Arizona State University, 2002.

[6] D. Puig, M.A. Garcia, Automatic texture feature selection for image pixel classification, Pattern Recognition 39 (11) (2006) 1996–2009.

[7] M.A. Garcia, D. Puig, Supervised texture classification by integration of multiple texture methods and evaluation windows, Image and Vision Computing 25 (7) (2007) 1091–1106.

[8] G. Smith, I. Burns, Measuring texture classification algorithms. (Meastex image texture database and test suite), Pattern Recognition Letters 18 (1997) 1495–1501.

[9] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (7) (2002) 971–987.

[10] G.H. John, R. Kohavi, K. Pfleger, Irrelevant features and the subset selection problem, in: Proceedings of the 11th International Conference on Machine Learning, New Brunswick NJ, USA, 1994, pp. 121–129.

[11] A.H. Bhalerao, N.M. Rajpoot, Discriminant feature selection for texture classification, in: British Machine Vision Conference, 2003.

[12] M.E. Farmer, A.K. Jain, A wrapper-based approach to image segmentation and classification, IEEE Transactions on Image Processing 14 (12) (2005) 2060–2072.

[13] H. Liu, R. Setiono, Feature selection and classification: a probabilistic wrapper approach, in: Proceedings of the 9th International Conference on Industrial and Engineering Applications of AI and ES, Fukuoka, Japan, 1996, pp. 419–424.

[14] D.B. Skalak, Prototype and feature selection by sampling and random mutation hill climbing algorithms, in: Proceedings of the 11th International Conference on Machine Learning, New Brunswick, NJ, 1994, pp. 293–301.

[15] L.C. Molina, L. Belanche, A. Nebot, Feature selection algorithms: a survey and experimental evaluation, in: International Conference on Data Mining, Japan, 2002, pp. 306–313.

[16] M.A. Garcia, D. Puig, Robust aggregation of expert opinions based on conflict analysis and resolution, Lecture Notes in Artificial Intelligence, CAEPIA, vol. 3040, Springer, Berlin, 2004, pp. 488–497.

[17] A.R. Rao, G.L. Lohse, Towards a texture naming system: identifying relevant dimensions of texture, Vision Research 36 (11) (1996) 1649–1669.

[18] P. Brodatz, Textures: A Photographic Album for Artists and Designers, Dover & Greer Publishing Company, 1999.

[19] H. Long, W.K. Leow, A hybrid model for invariant and perceptual texture mapping, in: Proceedings of the International Conference on Pattern Recognition, vol. 1, 2002, pp. 135–138.

[20] M. Vanrell, R. Baldrich, A. Salvatella, R. Benavente, F. Tous, Induction operators for a computational colour texture representation, Computer Vision and Image Understanding 94 (1–3) (2004) 92–114.

[21] M.N. Do, M. Vetterli, Wavelet-based texture retrieval using generalized gaussian density and Kullback–Leibler distance, IEEE Transactions on Image Processing 11 (2) (2002) 146–158.

[22] B. Balas, Attentive texture similarity as a categorization task: comparing texture synthesis models, Pattern Recognition 41 (3) (2008) 972–982.

[23] M. Mirmehdi, M. Petrou, Segmentation of color textures, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (2) (2000) 142–159.

[24] D. Depalov, T.N. Pappas, D. Li, B. Gandhi, Perceptual feature selection for semantic image classification, in: International Conference on Image Processing, Atlanta, USA, 2006, pp. 2921–2924.

[25] J.G. Leu, On indexing the periodicity of image textures, Image and Vision Computing 19 (13) (2001) 987–1000.

[26] T. Randen, J.H. Husoy, Filtering for texture classification: a comparative study, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (4) (1999) 291–310.

[27] MIT Vision and Modelling Group, 1998 〈http://www.media.mit.edu/vismod/〉.

[28] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: IEEE International Conference on Computer Vision, vol. II, 2001, pp. 416–423.

[29] B.S. Manjunath, W.Y. Ma, Texture features for browsing and image retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (8) (1996) 837–842.

[30] L. Chen, G. Lu, D. Zhang, Effects of different gabor filter parameters on image retrieval by texture, in: Proceedings of the 10th International Multimedia Modelling Conference, 2004, pp. 273–278.

[31] N. Bourbakis, R. Patil, A methodology for automatically detecting texture paths and patterns in images, in: Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence, Patras, Greece, vol. 1, 2007, pp. 504–512.

About the Author—DOMENEC 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.

About the Author—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.

About the Author—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 in the Intelligent Robotics and Computer Vision Group at Rovira i Virgili University. His research interests include pattern recognition, image processing, texture analysis and robotics.