Journal of Mathematical Imaging and Vision manuscript No.
(will be inserted by the editor)

Texture description through histograms of equivalent patterns

Antonio Fernández · Marcos X. Álvarez · Francesco Bianconi

Received: date / Accepted: date

Abstract The aim of this paper is to describe a general framework for texture analysis which we refer to as the HEP (histograms of equivalent patterns). The HEP, of which we give a clear and unambiguous mathematical definition, is based on partitioning the feature space associated to image patches of predefined shape and size. This task is approached by defining, a priori, suitable local or global functions of the pixels' intensities. In a comprehensive survey we show that diverse texture descriptors, such as co-occurrence matrices, gray-level differences and local binary patterns, can be seen all to be examples of the HEP. In the experimental part we comparatively evaluate a comprehensive set of these descriptors on an extensive texture classification task. Within the class of HEP schemes, improved local ternary patterns (ILTP) and completed local binary patterns (CLBP) emerge as the best of parametric and non-parametric methods, respectively. The results also show the following patterns: 1) higher effectiveness of multi-level discretization in comparison with binarization; 2) higher accuracy of parametric methods when compared to non-parametric ones; 3) a general trend of increasing performance with increasing dimensionality; and 4) better performance of point-to-average thresholding against point-to-point thresholding.

Keywords Image classification · Texture features · BGC · LBP · LTP

1 Introduction

Texture analysis is an area of intense research. The keen interest in this topic stems from the important role that it plays in many disciplines and related applications: computer-assisted diagnosis, remote sensing, surface grading, defect detection and food inspection are just some examples where texture analysis, is, nowadays, a standard. It is curious that such a widely used concept – *texture* – has not found a general consensus regarding an explicit definition. Even experienced researchers do not seem at ease when it comes to defining texture, preferring instead to characterize it by stating what it is and what it is not [18].

Whereas defining texture has proven quite challenging so far, many methods have been proposed to describe it in a quantitative way. So many, indeed, that the whole set has been recently referred to as 'a galaxy of texture features' [127]. Various authors have tried to put order to this galaxy through suitable classification schemes. A first subdivision was proposed by van Gool *et al.* into statistical and structural methods [27]. Later on Tuceryan and Jain [108] introduced a taxonomy in four categories (statistical, geometrical, model based and signal processing methods) which has received a great deal of attention in the computer vision community [78,132,100,101]. More recently Sonka

A. Fernández
School of Industrial Engineering, University of Vigo
Campus Universitario, 36310 Vigo, Spain
Tel.: 0034-986-818602
Fax: 0034-986-812201
E-mail: antfdez@uvigo.es

M. X. Álvarez
School of Mining Engineering, University of Vigo
E-mail: marcos@uvigo.es

F. Bianconi*
Department of Industrial Engineering, University of Perugia
Via G. Duranti, 67
06125 Perugia, Italy
E-mail: bianco@ieee.org
*Performed this work as a visiting researcher in the School of Industrial Engineering, University of Vigo, Spain

et al. described a classification framework based on statistical, syntactic and hybrid methods [104]. It has been argued that a crisp distinction among these categories could be not entirely satisfactory, since there are methods that present distinctive traits belonging to more than one class. Concerning this, Xie and Mirmehdi [127] suggested that the categories presented in Ref. [108] should rather be regarded as *attributes* that one specific method may possess or not. This 'fuzziness' could be related to the lack of a formal definition of the above categories, which are based on intuitive attributes (i.e. *statistical*, *structural*, etc.) rather than formal, mathematical definitions. The aim of this paper is, indeed, to investigate a class of texture descriptors that can be easily defined in a formal way, as we will show in Section 2.2, therefore generating no doubt whether a method pertains to this class or not. This class includes those methods that characterize a texture image through the probability of occurrence of the patterns associated to a neighbourhood of given size and shape. For reasons that will become clear shortly, we identify this class with the acronym *HEP* (Histograms of Equivalent Patterns). In the remainder of the paper we show that many renown methods such as texture spectrum, local binary patterns and co-occurrence matrices belong to the HEP, and can be regarded as variations of the same underlying idea. The proposed framework gives also a chance to elaborate on some basic concepts in texture analysis such as the dichotomy between unlearnt (*a priori*) vs. learnt (*a posteriori*) feature systems, an issue that we discuss in detail in Sec. 5.1.

Literature review shows that no attempts to present these methods in a comprehensive and unifying framework have been proposed hitherto. We deem nonetheless it opportune to mention two relevant works in which subsets of the methods studied herein are included in comparative reviews. Nanni *et al.* [77] compared the performance of some LBP-based methods on a set of classification tasks related to biomedical images. Likewise, Huang *et al.* [48] presented a survey of LBP-based approaches for facial image analysis. Both references are very recent, providing evidence for the high interest in this topic.

The remainder of the manuscript is organized as follows. After an introductory discussion about the underlying concepts (Sec. 2.1), we introduce the mathematics of the HEP in Sec. 2.2. In Sec. 3 we present a review of the methods that belong to the framework. The possible extensions of the framework are discussed in Sec. 4. In Sec. 5 we discuss about two fundamental issues in texture analysis (dichotomies *a priori* vs. *a posteriori* and image patches vs. filter responses) and put the HEP in the context of this debate. In Sec. 6 we present an extensive comparative experimental study. Final considerations (Sec. 7) conclude the paper.

2 Histograms of equivalent patterns

2.1 Basic concepts

There is a general consensus, in literature, about the fact that stationary texture images – i.e.: images containing a single type of texture [94] – can be conveniently characterized through the probability distribution of the possible grey-scale instances of a predefined neighbourhood [66]. These local features, usually referred to as *texels* or *textons*, are frequently repeated elements that make up the texture's structure. Orderless sets of such elements – sometimes referred to as 'bag-of-features' – have been proved to be highly descriptive (and predictive) of a certain texture class [131]. Their probability could be in principle estimated through a histogram that measures the frequency of occurrence of the different grey-scale patterns throughout the image. To compute such histogram, one makes the neighbourhood move by steps of one pixel across the image, and at each position increments the bin corresponding to the detected pattern by one unit. Although this approach results attractive for its conceptual simplicity, a straightforward application of the method is impractical, since the number of entries in the histogram would be overwhelmingly large, even for small neighbourhoods – if we were to take into account all the possible patterns. Considering that the typical depth of digitization of most commercial imaging devices is 256, the number of different grey-scale patterns for a 3×3 square neighbourhood would be $256^9 = 2^{72}$, which means a feature vector of approximately 4.7×10^{21} components. Since the number of possible patterns is several orders of magnitude greater than the number of image pixels, even for high resolution imagery, the vast majority of histogram bins would remain empty. Such an extremely sparse, ultra high-dimensional histogram would provide an unreliable estimation of the underlying distribution and have scarce – if any – discriminant power in image description [63]. Moreover, the amount of memory required to store one of such histograms would largely exceed the capacity of the currently available computers. The simplest way to reduce the joint histogram dimensionality would be decreasing the number of grey levels. This is not of great help, however, since the number of required bins gets significantly large even with few grey levels (a 3×3 neighbourhood would require $4^9 = 262144$ bins with as few as four grey levels). Any method belonging to the HEP deals with this problem by defining a partition of the pattern space into classes of equivalent patterns, and by merging the histogram bins of the equivalent ones. The number of classes K into which the pattern space is subdivided represents the dimensionality of the method (throughout the paper we assume that classes are labelled from 0 to $K-1$). The problem is therefore reduced to defining a suitable criterion for partitioning the pattern space. Several criteria have been proposed in the past. They represent an ample family of texture descriptors which share the same underlying principle, though they have been developed and presented independently in literature, and look rather unrelated. The methods of this family have a number of important advantages: they are conceptually simple, easy to implement and reasonably fast. Some of them are very popular in the computer vision community, such as LBP and related methods.

In the following section we restate the concepts summarized here in mathematical language. In particular we show that the whole family and each method belonging to it can be expressed precisely and unambiguously.

2.2 Formal definition

A mathematical definition of the HEP makes it possible to avoid the uncertainty that some of the classifying schemes proposed thus far entail.

First of all we introduce the notation to be used henceforth. Let $\mathbf{I}$ be an $M \times N$ matrix representing the raw pixel intensities of an image quantized to G grey levels ranging from 0 to $G-1$, and $I_{m,n}$ the grey-scale intensity at pixel (m,n).

Definition 1 A texture descriptor is a function F that receives an image $\mathbf{I}$ as input and returns a vector $\mathbf{h}$:

$$\mathbf{h} = F(\mathbf{I}) \tag{1}$$

where $\mathbf{h}$ is usually referred to as the *feature vector*.

Definition 2 Histograms of equivalent patterns (HEP) is a class of texture descriptors for which the k-th element of $\mathbf{h}$ can be expressed in the following way:

$$h_k = \frac{1}{D} \sum_{m=m_{min}}^{m_{max}} \sum_{n=n_{min}}^{n_{max}} \delta \left[f\left(\mathbf{x}_{m,n}^{\Omega}, \boldsymbol{\theta} \right) - k \right] \tag{2}$$

where m and n represent row- and column-wise pixel indices, $\mathbf{x}_{m,n}^{\Omega}$ the grey-scale values of a set of pixels defining a generic neighbourhood $\Omega_{m,n}$ around (m,n), $\boldsymbol{\theta}$ a vector of parameters computed from the whole image, D a normalizing factor, δ the function defined in Eq. 7 and f a generic function that returns an integer between 0 and $K-1$. The limits of the sums in Equation 2 are intended to guarantee that $\Omega_{m,n}$ be fully enclosed inside the image $\mathbf{I}$.

In plain words the definition is straightforward: a texture descriptor belonging to the HEP is based on a neighbourhood of predefined shape which is moved along the image by steps of one pixel. For each position, one among K predefined class labels is assigned to the neighbourhood, and the corresponding k-th component of $\mathbf{h}$ is incremented by $1/D$. Therefore the feature vector represents the probability of occurrence of each class (factor $1/D$ normalizes the feature vector to sum one). Most commonly $\Omega_{m,n}$ defines a square, rectangular or circular window, but other arrangements have also been proposed in literature [2, 74, 91–93]. It is worth spending a couple of words about the meaning of $\boldsymbol{\theta}$. This represents a vector of parameters computed from the whole image, such as, for instance, a global threshold for binarization. Therefore Eq. 2 typifies a *global* texture descriptor, namely a descriptor that is based on – or, in some way, employs – global features. If we drop the dependence on $\boldsymbol{\theta}$, we obtain a subclass of texture descriptors belonging to the HEP that rely on local features only: we refer to them as *local* texture descriptors.

The definition of a texture descriptor belonging to the HEP is therefore a matter of determining a suitable function f. We refer to it as the *kernel function*. We discuss this concept in detail here below. For the sake of simplicity – but without loss of generality – we restrict the discussion to 3×3 square neighbourhoods. We believe this decision is supported both by isotropy and easiness-of-implementation considerations, as well as by the fact that most applications reported in literature are based on this setting. Consider a generic image $\mathbf{I}$ and let $\mathbf{x}_{m,n}^{3\times3}$ be the set of grey-scale values of a 3×3 square neighbourhood centred at (m,n):

$$\mathbf{x}_{m,n}^{3\times3} = \begin{bmatrix} I_{m-1,n-1} & I_{m-1,n} & I_{m-1,n+1} \\ I_{m,n-1} & I_{m,n} & I_{m,n+1} \\ I_{m+1,n-1} & I_{m+1,n} & I_{m+1,n+1} \end{bmatrix} \tag{3}$$

In this case the parameters in Eq. 2 take the following values: $m_{min} = n_{min} = 2$, $m_{max} = M-1$, $n_{max} = N-1$ and $D = (M-2)(N-2)$.

Now let $\mathcal{M}_{3\times3,G}$ be the set of all the possible instances defined by Eq. 3 and $\#\mathcal{M}_{3\times3,G}$ its cardinality. We refer to this entity as the *pattern space*, since it represents all the possible grey-scale patterns associated to a predefined neighbourhood (3×3 window, in this case). In the remainder of this section we use the symbol $\mathbf{x}$ to indicate a generic pattern of this type. As mentioned in Sec. 1, describing a texture directly through the frequency of occurrence of the elements of $\mathcal{M}_{3\times3,G}$ would result in a feature vector of 256^9 components (assuming $G = 256$). The HEP deals with such a high dimensional feature space by partitioning $\mathcal{M}_{3\times3,G}$ into groups of equivalent patterns through f. This function, in fact, defines an equivalence relation $\sim$ in $\mathcal{M}_{3\times3,G}$ that acts as follows:

$$\mathbf{x}_1 \sim \mathbf{x}_2 \Leftrightarrow f(\mathbf{x}_1) = f(\mathbf{x}_2) \quad \forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{M}_{3\times3,G}. \tag{4}$$

This relation induces a partition in $\mathcal{M}_{3\times3,G}$ that can be expressed in the following way:

$$\mathcal{M}_{3\times3,G} = \bigcup_{0\leq k\leq K-1} \mathcal{M}_{f,k} \tag{5}$$

where the family of subsets $\{\mathcal{M}_{f,k} \mid 0 \leq k \leq K-1\}$ is pairwise disjoint, and each subset is defined by:

$$\mathcal{M}_{f,k} = \{\mathbf{x} \in \mathcal{M}_{3\times3,G} \mid f(\mathbf{x}) = k\} \tag{6}$$

If we consider a neighbourhood Ω of generic shape and size, the above reasoning equally holds: f still defines a partition of the pattern space $\mathcal{M}_{\Omega,G}$. In this case the number of possible patterns is G^ω, where ω is the number of pixels in Ω. In principle any function f defines a texture descriptor belonging to the HEP. In practice it is recommendable that f satisfy some reasonable constraints. A sensible criterion, for instance, could be that the induced equivalence relation be perceptually meaningful, or, in other words, that similar patterns be mapped into the same equivalence class. Another important condition is that f provide effective dimensionality reduction, i.e. $K \ll G^9$.

The concept of space partitioning, upon which are based not only the methods considered herein, but many others too, has gone rather unnoticed in literature. It was only recently, in fact, that Griffin *et al.* [29,16] did very clearly set into evidence this concept establishing a direct link between detection of local features and space partitioning. The main difference with the HEP – which deals directly with image patches without previous filtering – is that the method presented in Ref. [16] is based on partitioning the response space (also referred to as *jet space*) obtained transforming the original image through a set of Gaussian-derivative filters.

2.3 Combinations of mappings

In some cases, which will be considered in the remainder of the paper, a texture descriptor can be obtained combining two or more equivalence relations. The two combination approaches that we consider here are *concatenation* and *joint description*.

Let f_1 and f_2 be two mappings, and K_1 and K_2 the dimensions of the corresponding feature vectors. Concatenation generates a new feature vector that contains the elements of both f_1 and f_2, therefore its dimension is $K_1 + K_2$. We use the symbol $||$ to indicate this operation.

Joint description means that each class is uniquely identified by two labels, each one generated by a different mapping. Conceptually this operation is very similar to a Cartesian product, thus we indicate it with the symbol ×. The number of features is K_1K_2 in this case. In the implementation adopted here, this type of representation is serialized into a one-dimensional feature vector, with the convention that the $(k_1K_2 + k_2)$-th element corresponds to class labels k_1 and k_2 of, respectively, f_1 and f_2.

2.4 Geometrical interpretation

We have already pointed up that defining a texture descriptor belonging to the HEP is equivalent to partitioning an ω-dimensional space, where ω is the number of pixels in the neighbourhood. We can therefore reinterpret the overall problem from the viewpoint of high-dimensional geometry. It is interesting to notice that the methods published so far, which are described in detail in Sec. 3, define partitions that can be expressed through systems of linear equalities and inequalities in the variables $I_{m,n}$ (Eq. 3). Geometrically these equations represent polytopes in the ω-dimensional space. These are convex figures bounded by hyperplanes, and can be considered the analogue of polygons in two dimensions and polyhedra in three dimensions [103]. Polytopes have been studied intensely and most facts about them are now known, such as how to compute their volume and enumerate integer points in them [3]. Establishing a direct link between polytopes and texture descriptors makes it possible to derive theoretical properties of the latter by studying the corresponding polytopes, as suggested in [7].

3 Revisiting the existing methods

In the preceding section we stated that generating a texture descriptor belonging to the HEP is a matter of determining a function f which defines a suitable partition of the pattern space $\mathcal{M}_{\Omega,G}$. We present, in this section, a review of the main approaches reported in literature. As mentioned in Sec. 2.2, we limit the review to the $\mathcal{M}_{3\times3,G}$ pattern space. Extensions of the framework, including neighbourhoods of different shape and size, rotation invariance and other adjustments are discussed in Sec 4.

As a preliminary step we define four functions of the real variable x that are extensively used throughout the paper. These are:

– the δ function

$$\delta(x) = \begin{cases} 1, & \text{if } x = 0 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

– the binary thresholding function

$$\xi(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \tag{8}$$

– the ternary thresholding function

$$\eta_{\Delta}(x) = \begin{cases} 0, & \text{if } x < -\Delta \\ 1, & \text{if } -\Delta \leq x \leq \Delta \\ 2, & \text{if } x > \Delta \end{cases} \tag{9}$$

– the quinary thresholding function

$$\psi_{\Delta_1,\Delta_2}(x) = \begin{cases} 2, & \text{if } x \geq \Delta_2 \\ 1, & \text{if } \Delta_1 \leq x < \Delta_2 \\ 0, & \text{if } -\Delta_1 \leq x < \Delta_1 \\ -1, & \text{if } -\Delta_2 \leq x < -\Delta_1 \\ -2, & \text{if } x < -\Delta_2 \end{cases} \tag{10}$$

where Δ, Δ_1 and Δ_2 are positive real numbers; $\Delta_1 < \Delta_2$.

Next, for the sake of simplicity, let us take out the sub-indices m, n of Eq. 3 and indicate the set of greyscale values of a 3×3 neighbourhood in a more manageable way:

$$\mathbf{x} = \begin{bmatrix} I_7 & I_6 & I_5 \\ I_0 & I_c & I_4 \\ I_1 & I_2 & I_3 \end{bmatrix} \tag{11}$$

In the above equation I_c is the grey level of the central pixel and I_j the grey levels of the peripheral pixels ($j \in \{0, 1, \ldots, 7\}$).

We are now ready to start with the list of methods belonging to the HEP. The methods are presented in chronological order and summarized in Tab. 1. Of each technique we briefly recall the basics and provide the mathematical formulation within the HEP by defining f.

3.1 Local methods

3.1.1 Grey level co-occurrence matrices

Haralick's grey level co-occurrence matrices (GLCM) [34] measure the joint probability of the grey-levels of two pixels standing in a predefined relative position. Within the 3×3 window the standard approach [34] considers the four one-pixel displacements corresponding to directions 0°, 45°, 90° and 135°. The kernel function can therefore be written as follows:

$$f_{\mathrm{GLCM},j}(\mathbf{x}) = GI_c + I_j \tag{12}$$

where $j \in \{4, \cdots, 7\}$. The feature vector $\mathbf{h}$ is obtained by averaging the four vectors $\mathbf{h}_{\mathrm{GLCM},j}$ corresponding to each direction:

$$\mathbf{h}_{\mathrm{GLCM}} = \frac{1}{4}\sum_{j=4}^{7} \mathbf{h}_{\mathrm{GLCM},j} \tag{13}$$

Note that Eq. 12 serializes what it is usually referred to as the "co-occurrence matrix" into a vector whose $(GI_c + I_j)$-th element corresponds to the (I_c, I_j) element of the matrix. In the HEP the co-occurrence coefficients defined by Eq. 12 are used directly as texture features, as proposed in Refs. [116,89]. The resulting dimension is therefore G^2. Alternatively one can use synthetic statistical descriptors (i.e.: energy, entropy, correlation, contrast, etc.) extracted from the co-occurrence coefficients [94,12,28,79].

3.1.2 Gray level differences

Gray level differences (GLD) [120] are based on the probability distribution of the absolute difference between the grey levels of two pixels standing in a predefined relative position. The kernel function in this case is:

$$f_{\mathrm{GLD},j}(\mathbf{x}) = |I_c - I_j| \tag{14}$$

Similarly to GLCM, the feature vector is obtained by averaging the four vectors $\mathbf{h}_{\mathrm{GLD},j}$ corresponding to the orientations 0°, 45°, 90° and 135°:

$$\mathbf{h}_{\mathrm{GLD}} = \frac{1}{4}\sum_{j=4}^{7} \mathbf{h}_{\mathrm{GLD},j} \tag{15}$$

Table 1: Summary of texture descriptors belonging to the HEP.

Name	Acronym	Dim.	Ker. fun.	Ref.	Year
Local methods					
Grey level co-occurrence matrices	GLCM	G^2	Eq. 12	[34]	1973
Grey level differences	GLD	G	Eq. 14	[120]	1976
Sum and difference histograms	SDH	$2(2G-1)$	Eqs. 16,17	[110]	1986
Texture spectrum (0)	TS0	3^8	Eq. 20	[38]	1990
Texture spectrum ($\varDelta$)	TS$\varDelta$	3^8	Eq. 21	[40]	1992
Rank transform	RT	9	Eq. 22	[129]	1994
Reduced texture units	RTU	45	Eq. 27	[58]	1995
Gray level texture co-occurrence spectrum	GLTCS+	4!	Eq. 28	[45]	1996
Local binary patterns	LBP, CLBP_S	2^8	Eq. 29	[82]	1996
Simplified texture spectrum	STS	3^4	Eq. 30	[128]	2003
Simplified texture units (+)	STU+	3^4	Eq. 31	[71]	2003
Simplified texture units (×)	STU×	3^4	Eq. 32	[71]	2003
Modified texture spectrum	MTS	2^4	Eq. 33	[128]	2003
Improved local binary patterns	ILBP	2^9-1	Eq. 34	[50]	2004
Gradient texture unit coding	GTUC	$2 \cdot 3^7$	Eq. 37	[13]	2004
3D Local Binary Patterns	3DLBP	$4 \cdot 2^8$	Eqs. 29,38	[46]	2006
Center-symmetric local binary patterns	CS-LBP	2^4	Eq. 40	[42]	2006
Median binary patterns	MBP	2^9-1	Eq. 41	[33]	2007
Local ternary patterns	LTP	$2 \cdot 2^8$	Eqs. 44,45	[107]	2007
Centralized binary patterns	CBP	2^5	Eq. 47	[24]	2008
Improved center-symmetric local binary patterns (D)	D-LBP	2^4	Eq. 49	[126]	2009
Improved center-symmetric local binary patterns (ID)	ID-LBP	2^4	Eq. 50	[126]	2009
Improved local ternary patterns	ILTP	$2 \cdot 2^9$	Eqs. 52,53	[75]	2010
Local quinary patterns	LQP	$4 \cdot 2^8$	Eq. 56	[74]	2010
Binary gradient contours (1)	BGC1	2^8-1	Eq. 58	[22]	2011
Binary gradient contours (2)	BGC2	2^8-1	Eq. 59	[22]	2011
Binary gradient contours (3)	BGC3	$(2^4-1)^2$	Eq. 60	[22]	2011
Center-symmetric texture spectrum	CS-TS$\varDelta$	3^4	Eq. 61	[130]	2011
Improved center-symmetric texture spectrum	ICS-TS$\varDelta$	$2 \cdot 3^2$	Eqs. 62,63	[130]	2011
Gradient-based local binary patterns	GLBP	2^8	Eq. 65	[41]	2011
Improved binary gradient contours (1)	IBGC1	$2 \cdot (2^8-1)$	Eq. 67	This paper	
Global methods					
Binary texture co-occurrence spectrum	BTCS+	2^4	Eq. 68	[90]	1991
Coordinated clusters representation	CCR	2^9	Eq. 69	[61]	1996
Completed local binary patterns (C)	CLBP_C	2	Eq. 70	[31]	2010
Completed local binary patterns (M)	CLBP_M	2^8	Eq. 72	[31]	2010

3.1.3 Sum and difference histograms

Sum and difference histograms [110] are conceptually similar to GLD, but, instead of considering the absolute difference of two grey-scale values, they take into account either their sum (sum histogram – SH) or difference (difference histogram – DH). The kernel functions are:

$$f_{\mathrm{SH},j}(\mathbf{x}) = I_c + I_j \tag{16}$$

$$f_{\mathrm{DH},j}(\mathbf{x}) = (I_c - I_j) + (G - 1) \tag{17}$$

for sum and difference histogram respectively. The term $(G-1)$ in Eq. 17 guarantees that the function returns non negative integers. The two resulting feature vectors are concatenated into a descriptor referred to as sum and difference histograms (SDH):

$$\mathbf{h}_{\mathrm{SDH},j} = \mathbf{h}_{\mathrm{DH},j} || \mathbf{h}_{\mathrm{SH},j} \tag{18}$$

Finally, the concatenation of the four histograms corresponding to orientations 0°, 45°, 90° and 135° leads to the implementation proposed in Ref. [110], which is the one adopted here:

$$\mathbf{h}_{\mathrm{SDH}} = \mathbf{h}_{\mathrm{SDH},4} || \mathbf{h}_{\mathrm{SDH},5} || \mathbf{h}_{\mathrm{SDH},6} || \mathbf{h}_{\mathrm{SDH},7} \tag{19}$$

3.1.4 Texture spectrum

Texture spectrum, introduced by He and Wang [38], can be considered the precursor of a set of more recent methods such as local binary patterns and the like. In its original formulation it is based on the ternary thresholding function (Eq. 9) with $\varDelta = 0$. We refer to this method as TS0. In this model each peripheral pixel of the 3×3 neighbourhood is assigned a value 0, 1 or 2 if its grey-level intensity is less, equal or greater than the intensity of the central pixel, respectively. This defines a set of 3^8 possible ternary patterns. The corresponding kernel function is:

$$f_{\mathrm{TS0}}(\mathbf{x}) = \sum_{j=0}^{7} 3^j \eta_0 \left(I_j - I_c\right) \tag{20}$$

Later on the same authors described a variation of the method where $\varDelta$ takes a value different than zero [40]. This improvement should be potentially beneficial in presence of noise, since a grey-level variation below $\varDelta$ does not change a ternary pattern into another. We indicate this method with the acronym TS$\varDelta$, and the corresponding kernel function is formally analogous to Eq. 20:

$$f_{\mathrm{TS}\varDelta}(\mathbf{x}) = \sum_{j=0}^{7} 3^j \eta_{\varDelta} \left(I_j - I_c\right) \tag{21}$$

3.1.5 Rank transform

The rank transform (RT) [129] takes into account the number of pixels in the periphery of the 3×3 region the intensity of which is less than the intensity of the central pixel. Since this number ranges from zero to eight, this results in a set of nine possible patterns. The kernel function is:

$$f_{\mathrm{RT}}(\mathbf{x}) = \sum_{j=0}^{7} \xi \left(I_c - I_j - 1\right) \tag{22}$$

3.1.6 Reduced texture units

Reduced texture units (RTU) [58] can be viewed as a compressed version of texture spectrum which considers only the total number of 0s, 1s and 2s that appear in the periphery of the 3×3 neighbourhood after the action of the ternary thresholding function η_0. It is convenient to observe that the method generates as many features as the number of non-negative integer solutions to the equation $\alpha_0 + \alpha_1 + \alpha_2 = 8$, where α_0, α_1 and α_2 are the number of 0s, 1s and 2s, respectively. This problem is the same as counting the number of ways w through which n objects can be placed into r distinct cells, which is usually referred to as the *occupancy problem*. The solution turns out to be [20]:

$$w = \binom{n + r - 1}{r - 1} \tag{23}$$

In this case, since $n = 8$ and $r = 3$, the number of features which result is 45. Now, let α_0 and α_1 be the number of 0s and 1s in the periphery after ternary thresholding:

$$\alpha_0 = \sum_{j=0}^{7} \delta \left[\eta_0 \left(I_j - I_c\right)\right] \tag{24}$$

$$\alpha_1 = \sum_{j=0}^{7} \delta \left(I_j - I_c\right) \tag{25}$$

In order to derive a compact expression for the kernel function, let us observe that this can be viewed as a way to encoding all the possible couples (α_0, α_1) which satisfy the constraint $\alpha_0 + \alpha_1 \leq 8$. Geometrically this is equivalent to listing all the lattice points in the 2-simplex of vertices $(0, 0)$, $(8, 0)$ and $(0, 8)$, a problem which leads to the following expression:

$$f_{\mathrm{RTU}}(\mathbf{x}) = \alpha_0 + \sum_{r=\alpha_1}^{7} \sum_{s=r}^{7} 1 \tag{26}$$

We can easily compute the double sum in the above equation and give the kernel function the following simple form:

$$f_{\mathrm{RTU}}(\mathbf{x}) = \alpha_0 + \frac{\left(8 - \alpha_1\right)\left(9 - \alpha_1\right)}{2} \tag{27}$$

3.1.7 Gray level texture co-occurrence spectrum

The gray level texture co-occurrence spectrum [45] considers the probability of occurrence of the possible states (orderings) that arise when the pixels of the neighbourhood are sorted in descending order by their grey-scale value. Given a neighbourhood of predefined size and shape, the number of possible orderings is the number of permutations of ω objects, where ω is the number of pixels of the neighbourhood. Consequently the approach generates $\omega!$ features. In the implementation adopted here, which we refer to as GLCTS+, we considered the neighbourhood formed by pixels 0, 2, 4 and 6. The mathematical formulation that we propose here is based on Lehmer's code [1]:

$$f_{\mathrm{GLCTS+}}(\mathbf{x}) = \sum_{i=1}^{3} \sum_{j=i+1}^{4} i!\ \xi[I_{\pi^{-1}(j)} - I_{\pi^{-1}(i)}] \tag{28}$$

where $\pi^{-1}(k)$ represents the index of the pixel whose intensity occupies the k-th position in the list sorted in ascending order. In this case positions range from 1 to 4.

3.1.8 Local binary patterns

Local binary patterns (LBP) have received a great deal of attention in the pattern recognition community [65]. In the 3×3 domain the LBP operator thresholds the eight peripheral pixels of the neighbourhood at the value of the central pixel, thus defining a set of 2^8 possible binary patterns. The kernel function is:

$$f_{\mathrm{LBP}}(\mathbf{x}) = \sum_{j=0}^{7} 2^j \xi \left(I_j - I_c\right) \tag{29}$$

It is convenient at this point to digress a bit about the origin of this very influential method. Related literature has always had it that LBP was developed within the Machine Vision Group at the University of Oulu, Finland. Perhaps this belief relies on a technical report [35], where members of the group mention a forthcoming article about the method. However, to the best of our knowledge, this article has never been published. In contrast, the first two articles that explicitly mention LBP were published one year later [81,95]. Later on, in 1996, appeared the article that it is usually considered the origin of the method [82]. But Zabih and Woodfill [129] had already presented a method, referred to as census transform, which resembles the LBP to a great extent, the only difference being the verse of the inequality in Eq. 8. Finally, and surprisingly enough, we found that the fundamental idea was already detailed in a work of Gong *et al.* [26], where LBP is a by-product of a simplification of texture spectrum.

3.1.9 Simplified texture spectrum

Texture spectrum has been the basis of a good number of variations, all aiming to reduce the rather high dimensionality of the method. Xu *et al.* [128] proposed a simplified version which we indicate here as simplified texture spectrum (STS). Their method is based on the observation that, as the neighbourhood moves across the image at steps of one pixel, a generic couple of pixels (c, j) switches into its symmetric, i.e.: (j, c). Consequently each comparison between the central pixel and each pixel in the periphery is performed twice. In order to avoid this redundancy and reduce complexity, the authors consider the neighbourhood formed by the central pixel and the four adjacent peripheral pixels of one quadrant only, specifically those corresponding to indices $j \in \{4, \cdots, 7\}$ (see Eq. 11). This setting reduces the number of features from 3^8 to 3^4. The kernel function can be expressed as follows:

$$f_{\mathrm{STS}}(\mathbf{x}) = \sum_{j=4}^{7} 3^{(j-4)} \eta_0 \left(I_c - I_j\right) \tag{30}$$

3.1.10 Simplified texture units

Madrid-Cuevas *et al.* [71] proposed two simplified versions of texture spectrum referred to as simplified texture units. The first one, STU+, considers the neighbourhood composed of the central pixel and its vertically- and horizontally-connected peripheral pixels (i.e.: pixels 0, 2, 4 and 6 – Eq. 11). The second one, STU×, operates on the neighbourhood formed by the central pixel and its diagonally-connected peripheral pixels (i.e.: pixels 1, 3, 5 and 7 – Eq. 11). In both cases dimensionality is reduced from 3^8 to 3^4. The corresponding kernel functions are:

$$f_{\mathrm{STU+}}(\mathbf{x}) = \sum_{j=0}^{3} 3^j \eta_\Delta \left(I_{2j} - I_c\right) \tag{31}$$

$$f_{\mathrm{STU}\times}(\mathbf{x}) = \sum_{j=0}^{3} 3^j \eta_\Delta \left(I_{2j+1} - I_c\right) \tag{32}$$

3.1.11 Modified texture spectrum

Modified texture spectrum (MTS) can be considered as a simplified version of LBP, where only a subset of the peripheral pixels (i.e. pixels 4, 5, 6 and 7 – Eq. 11) is considered. To be precise the inequality in ξ is flipped in the original formulation of MTS [128], but this unimportant difference in no way alters the information that the method conveys. The kernel function of is:

$$f_{\mathrm{MTS}}(\mathbf{x}) = \sum_{j=4}^{7} 2^{(j-4)} \xi \left(I_c - I_j\right) \tag{33}$$

3.1.12 Improved local binary patterns

Improved local binary patterns (ILBP) are based on an idea similar to LBP, the only difference is that the whole 3×3 neighbourhood is thresholded by its average grey-scale value [50,51]. This gives $(2^9 - 1)$ possible binary patterns (the all 0s pattern is not possible by definition, hence the -1). The kernel function is:

$$f_{\mathrm{ILBP}}(\mathbf{x}) = 2^8 \xi \left(I_c - \bar{S}\right) + \sum_{j=0}^{7} 2^j \xi \left(I_j - \bar{S}\right) - 1 \tag{34}$$

where $\bar{S}$ is the average grey-scale value over the neighbourhood:

$$\bar{S} = \frac{1}{9}\left(I_c + \sum_{j=0}^{7} I_j\right) \tag{35}$$

We have already mentioned, in Sec. 3.1.8, that LBP is equivalent to the census transform, the only difference being the sense of inequalities in Eq. 8. Same thing occurs with ILBP, which is equivalent to the modified census transform [23].

3.1.13 Gradient texture unit coding

Gradient texture unit coding (GTUC) [13] considers the joint gradient computed along two vectors whose initial point is the central pixel of the 3×3 window, and whose terminal point is one of the peripheral pixels. To visualize the idea, let j_1 and j_2 be the indices of the terminal points of two vectors starting from the central pixel. Now consider the following quantity:

$$\varphi_{\Delta}(j_1, j_2) = \begin{cases} 0, & \text{if } |I_{j_1} - I_c| \leq \Delta \text{ and } |I_{j_2} - I_c| \leq \Delta \\ 1, & \text{if } |I_{j_1} - I_c| < \Delta \text{ and } |I_{j_2} - I_c| \geq \Delta \\ 1, & \text{if } |I_{j_1} - I_c| \geq \Delta \text{ and } |I_{j_2} - I_c| < \Delta \\ 2, & \text{if } |I_{j_1} - I_c| > \Delta \text{ and } |I_{j_2} - I_c| > \Delta \end{cases} \tag{36}$$

which is referred to as *gradient texture feature number* (GTFN). If we pin down j_1 (let, for instance, $j_1 = 7$) and make j_2 run circularly from 0 to 7, we obtain the GTFN corresponding to a relative angle of 45°, 90°, 135°, 180°, 225°, 270°, 315° and 0°, respectively. The authors correctly point out that what really counts in this setting is the relative angle between the two vectors, and that the choice of the first vector is arbitrary. Therefore the combination of the eight angles and the three possible values of the GTFN generates $2 \cdot 3^7$ possible patterns. Taking j_7 as the terminal point of the first vector, the kernel function can be formalized in the following way:

$$\begin{aligned} f_{\text{GTUC}}(\mathbf{x}) = 3^7 \delta \left[\varphi_{\Delta}\left(I_7, I_7\right) - 2\right] \\ + \sum_{j=0}^{6} 3^j \varphi_{\Delta}\left(I_j, I_7\right) \end{aligned} \tag{37}$$

3.1.14 3D Local Binary Patterns

3D Local Binary Patterns [46] (later on generalized to a multi-scale version referred to as Extended Local Binary Patterns [47]) are an improvement on LBP aiming to consider not only the sign, but also the value of the difference between the central pixel and the peripheral pixels. Therefore the method can be considered, to a certain extent, a combination of LBP and GLD. In devising the method, the authors moved from the consideration that the signed difference between the grey-scale value of the central pixel and that of a peripheral pixel can be represented through nine bits: one for the sign and the others for the absolute difference. They refer to these bits as *layers*. Now suppose that layer 0 represent the sign layer and layers from 1 to 8 the bits of the absolute difference, with the convention that layer 1 corresponds to the least significant bit. Each layer is a string of binary digits, and can be viewed as a binary pattern. Layer 0 is, in fact, the local binary pattern. Layers for 1 to 8 can be interpreted as higher-order binary patterns. The idea 3DLBP relies upon is to retain the three least significant bits (i.e.: layers 1, 2 and 3) and discard the others. All the absolute differences greater than 7 are assigned to 7. Considering that the method generates 2^8 features for each layer, the resulting dimension is $2^8 \cdot 4$. For a layer l the kernel function can be expressed as follows:

$$f_{\text{3DLBP},l}(\mathbf{x}) = \sum_{j=0}^{7} 2^j \beta_l\left(|I_j - I_c|\right) \tag{38}$$

where $\beta_l(x)$ is a function that returns the l-th binary digit of the decimal number x. The feature vector is obtained through concatenation of the vectors of each layer:

$$\mathbf{h}_{\text{3DLBP}} = \mathbf{h}_{\text{LBP}} || \mathbf{h}_{\text{3DLBP},1} || \mathbf{h}_{\text{3DLBP},2} || \mathbf{h}_{\text{3DLBP},3} \tag{39}$$

3.1.15 Center-symmetric local binary patterns

Center-symmetric local binary patterns (CS-LBP) [42] are similar to LBP, but employ a different scheme to compare the pixel in the neighbourhood. Whereas the central pixel plays a pivotal role in LBP, CS-LBP discards it altogether and considers the following centre-symmetric couples of pixel values (Eq. 11): (I_0, I_4), (I_1, I_5), (I_2, I_6) and (I_3, I_7). Robustness on flat image regions is obtained by thresholding the gray level differences with a parameter Δ. This generates a set of 2^4 possible patterns. The kernel function can be expressed in the following way:

$$f_{\text{CS-LBP}}(\mathbf{x}) = \sum_{j=0}^{3} 2^j \xi \left(I_j - I_{j+4} - \mathit{\Delta} - 1\right) \tag{40}$$

3.1.16 Median binary patterns

Median binary patterns (MBP) [33] have much in common with ILBP, the only difference is that MBP uses the median of the grey-scale values of the 3 × 3 neighbourhood as threshold, instead of the average value. The kernel function can be expressed as follows:

$$f_{\text{MBP}}(\mathbf{x}) = 2^8 \xi \left(I_c - \upsilon\right) + \sum_{j=0}^{7} 2^j \xi \left(I_j - \upsilon\right) - 1 \tag{41}$$

where υ is the median of the grey-scale values in neighbourhood.

3.1.17 Local ternary patterns

Local ternary patterns [107] can be considered a hybrid between texture spectrum and local binary patterns. Similarily to texture spectrum, in fact, they make use of the ternary thresholding function to obtain ternary binary patterns. Then each ternary pattern is split in two binary patterns (lower and upper) through the following rules:

$$b_{j,\text{LOWER}} = \begin{cases} 1, & \text{if } t_j = 0 \\ 0, & \text{otherwise} \end{cases} \tag{42}$$

$$b_{j,\text{UPPER}} = \begin{cases} 1, & \text{if } t_j = 2 \\ 0, & \text{otherwise} \end{cases} \tag{43}$$

where t_j and b_j represent, respectively, the ternary and binary value corresponding to pixel j. With this convention the kernel functions that define the two distributions (local ternary patterns lower – LTPL – and upper – LTPU) can be expressed in the following way:

$$f_{\text{LTPL}}(\mathbf{x}) = \sum_{j=0}^{7} 2^j \xi \left(I_c - I_j - \mathit{\Delta}\right) \tag{44}$$

$$f_{\text{LTPU}}(\mathbf{x}) = \sum_{j=0}^{7} 2^j \xi \left(I_j - I_c - \mathit{\Delta}\right) \tag{45}$$

The two descriptors are finally concatenated to form the LTP model:

$$\mathbf{h}_{\text{LTP}} = \mathbf{h}_{\text{LTPU}} || \mathbf{h}_{\text{LTPL}} \tag{46}$$

3.1.18 Centralized binary patterns

Centralized binary patterns (CBP) [24] consider the same couples of centre-symmetric pixels used by CS-LBP plus the central pixel. Relative comparison is based on the absolute difference of grey-scale values, which is thresholded at a predefined small positive value $\mathit{\Delta}$. The kernel function can be formalized as follows:

$$f_{\text{CBP}}(\mathbf{x}) = 2^4 \xi \left(\left|I_c - \bar{S}\right| - \mathit{\Delta}\right) + \sum_{j=0}^{3} 2^j \xi \left(\left|I_j - I_{j+4}\right| - \mathit{\Delta}\right) \tag{47}$$

where $\bar{S}$ is the average grey-scale value over the neighbourhood (Eq. 35).

3.1.19 Improved center-symmetric local binary patterns

In an effort to include the value of the central pixel in the model, a group of researchers of the Henan Polytechic University [126,52] proposed two variations on CS-LBP. In order to give a compact mathematical formulation of the two methods it is convenient to define, beforehand, the following function:

$$\phi\left(x_1, x_2, x_3\right) = \begin{cases} 1, & \text{if } x_1 \geq x_2 \text{ and } x_2 \geq x_3 \\ 1, & \text{if } x_1 < x_2 \text{ and } x_2 < x_3 \\ 0, & \text{otherwise} \end{cases} \tag{48}$$

The first method, originally referred to as ICS-LBP [126], and, later on, as D-LBP [52], considers the same four couples of centre-symmetric pixels as CS-LBP (Sec. 3.1.15), but includes the value of the central pixel in the computation. In practice the method takes into account the four triplets corresponding to the vertical and horizontal directions, and the two diagonal directions. Now let (i, c, j) be one of such triplets: the binary value corresponding to it is given by $\phi(I_i, I_c, I_j)$. Consequently the kernel function takes the following simple form:

$$f_{\text{D-LBP}}(\mathbf{x}) = \sum_{j=0}^{3} 2^j \phi \left(I_j, I_c, I_{j+4}\right) \tag{49}$$

The second method, denoted as ID-LBP [52], is actually very similar to the former, the only difference being that the value of the central pixel is replaced by the average value of the eight peripheral pixels. Therefore we have the following kernel function:

$$f_{\text{ID-LBP}}(\mathbf{x}) = \sum_{j=0}^{3} 2^j \phi \left(I_j, \hat{S}, I_{j+4}\right) \tag{50}$$

where $\hat{S}$ is the average grey-scale value over the eight peripheral pixels:

$$\hat{S} = \frac{1}{8}\sum_{j=0}^{7} I_j \tag{51}$$

3.1.20 Improved local ternary patterns

Improved local ternary patterns (ILTP) [75] are an extension of LTP where each pixel in the neighbourhood is thresholded at the average grey-scale value. Similarly to LTP the representation is split into a lower and upper part:

$$f_{\mathrm{ILTPL}}(\mathbf{x}) = 2^8\xi(\bar{S} - I_c - \Delta) + \sum_{j=0}^{7} 2^j \xi\left(\bar{S} - I_j - \Delta\right) \tag{52}$$

$$f_{\mathrm{ILTPU}}(\mathbf{x}) = 2^8\xi(I_c - \bar{S} - \Delta) + \sum_{j=0}^{7} 2^j \xi\left(I_j - \bar{S} - \Delta\right) \tag{53}$$

where $\bar{S}$ is the average value of the whole 3×3 neighbourhood (Eq. 35). The two descriptors are concatenated to give the ILTP model:

$$\mathbf{h}_{\mathrm{ILTP}} = \mathbf{h}_{\mathrm{ILTPH}} || \mathbf{h}_{\mathrm{ILTPL}} \tag{54}$$

3.1.21 Completed local binary patterns (CLBP_S)

Completed Local Binary Patterns (CLBP) have been recently introduced by Guo *et al.* [31] as an extension of local binary patterns. The approach is based on different combinations of three basic descriptors: CLBP_S, CLBP_C and CLBP_M. The first (a local method) is described here, whereas the other two (global methods) are described in Sec. 3.2.3. As for CLBP_S, this is just an alias for LBP, which we already described in Sec. 3.1.8.

3.1.22 Local quinary patterns

In local quinary patterns (LQP) [74] the grey level difference between the central pixel and the pixels of the periphery is encoded using five levels (i.e.: −2, −1, 0, 1 and 2) which are computed using two thresholds: Δ_1 and Δ_2. LQP is therefore closely related to LTP, the only difference being that the number of encoding levels is five in LQP and three in LTP. The quinary pattern is split into four binary patterns through the following rule:

$$b_{j,i} = \begin{cases} 1, & \text{if } q_j = i \\ 0, & \text{otherwise} \end{cases} \tag{55}$$

where $b_{j,i}$ and q_j are the binary and quinary value corresponding to pixel j and level i; $i \in \{-2, -1, 1, 2\}$. The kernel function corresponding to each level can be written as follows:

$$f_{\mathrm{LQP},i}(\mathbf{x}) = \sum_{j=0}^{7} 2^j \delta\left[\psi_{\Delta_1,\Delta_2}\left(I_c - I_j\right) - i\right] \tag{56}$$

The feature vector is obtained as follows:

$$\mathbf{h}_{\mathrm{LQP}} = \mathbf{h}_{\mathrm{LQP},-2} || \mathbf{h}_{\mathrm{LQP},-1} || \mathbf{h}_{\mathrm{LQP},1} || \mathbf{h}_{\mathrm{LQP},2} \tag{57}$$

3.1.23 Binary gradient contours

Binary gradient contours (BGC), recently introduced by the authors in Ref. [22], are a family of descriptors based on pairwise comparison of adjacent pixels belonging to one or more closed paths traced along the periphery of the 3×3 neighbourhood (hence the name *contours*). For each closed path a binary pattern is obtained by assigning each pair of adjacent pixels (i, j) in the path the binary value $\xi(I_j - I_i)$. Since there are several paths that one can pick out from the 3×3 neighbourhood, different BGC operators exist. In Ref. [22] we proposed three different operators, which are referred to as BGC1, BGC2 and BGC3. Both BCG1 and BCG3 are based on one closed path. In the first case the the pixels that define the path are: $\{0, 1, \cdots, 7, 0\}$ (Eq. 11), therefore the corresponding couples from which the binary values are extracted are: $\{(0, 1), (1, 2), \cdots, (7, 0)\}$. In the second case the path is defined by the following sequence of pixels: $\{0, 5, 2, 7, 4, 1, 6, 3, 0\}$ (couples are defined in the same way). Both descriptors generate $(2^8 - 1)$ possible different patterns, since, as it happens with ILBP (Sec. 3.1.12), the all-0s pattern is, by definition, impossible. In contrast BGC2 employs two closed paths, which are: $\{1, 7, 5, 3, 1\}$ and $\{0, 6, 4, 2, 0\}$. In this case each path generates $(2^4 - 1)$ possible patterns, therefore the combination of the two gives $(2^4 - 1)^2 = 225$ possible patterns. The kernel functions of the three models are reported here below:

$$f_{\mathrm{BGC1}}(\mathbf{x}) = \sum_{j=0}^{7} 2^j \xi\left(I_j - I_{(j+1) \bmod 8}\right) - 1 \tag{58}$$

$$f_{\mathrm{BGC2}}(\mathbf{x}) = (2^4 - 1)\sum_{j=0}^{3} 2^j \xi(I_{2j} - I_{2(j+1) \bmod 8}) \\ + \sum_{j=0}^{3} 2^j \xi(I_{2j+1} - I_{(2j+3) \bmod 8}) - 2^4 \tag{59}$$

$$f_{\mathrm{BGC3}}(\mathbf{x}) = \sum_{j=0}^{7} 2^j \xi(I_{3j \bmod 8} - I_{3(j+1) \bmod 8}) - 1 \tag{60}$$

3.1.24 Center-symmetric texture spectrum

Center-symmetric texture spectrum (CS-TSΔ) [130] can be viewed as a variation of texture spectrum (Sec. 3.1.4) in which the scheme to compare the pixels in the neighbourhood is the same as in CS-LBP (Sec. 3.1.15). It is worth mentioning that in the original reference [130] the method is indicated as center-symmetric local ternary patterns (CS-LTP). This is, however, quite a misleading name, since the mathematical formulation can be easily derived from TSΔ, but not from LTP (Sec. 3.1.17). To avoid this potential source of confusion, herein the method is referred to as center-symmetric texture spectrum. The kernel function is:

$$f_{\mathrm{CS\text{-}TS}\Delta}(\mathbf{x}) = \sum_{j=0}^{3} 3^j \eta_\Delta \left(I_j - I_{j+4}\right) \tag{61}$$

3.1.25 Improved center-symmetric texture spectrum

The improved center-symmetric texture spectrum (ICS-TSΔ) [130] is obtained by splitting CS-TSΔ in two parts: CS-TS1Δ and CS-TS2Δ. The former considers the couples (I_0, I_4) and (I_2, I_6), the latter the couples (I_1, I_5) and (I_3, I_7). Therefore the two kernel functions are:

$$f_{\mathrm{ICS\text{-}TS1}\Delta}(\mathbf{x}) = \sum_{j=0}^{1} 3^j \eta_\Delta \left(I_{2j} - I_{2j+4}\right) \tag{62}$$

$$f_{\mathrm{ICS\text{-}TS2}\Delta}(\mathbf{x}) = \sum_{j=0}^{1} 3^j \eta_\Delta \left(I_{2j+1} - I_{2j+5}\right) \tag{63}$$

The feature vector is obtained through concatenation of the feature vectors of CS-TS1Δ and CS-TS2Δ:

$$\mathbf{h}_{\mathrm{ICS\text{-}TS}\Delta} = \mathbf{h}_{\mathrm{ICS\text{-}TS1}\Delta} || \mathbf{h}_{\mathrm{ICS\text{-}TS2}\Delta} \tag{64}$$

3.1.26 Gradient-based local binary patterns

Another LBP-related method is represented by gradient-based local binary patterns (GLBP) [41]. Here the absolute difference between the central pixel and each peripheral pixel is thresholded at the mean absolute difference between (I_0, I_4) and (I_2, I_6). In formulas:

$$f_{\mathrm{GLBP}}(\mathbf{x}) = \sum_{j=0}^{7} 2^j \xi \left(I_+ - |I_j - I_c|\right) \tag{65}$$

where:

$$I_+ = \frac{1}{2}\left(|I_0 - I_4| + |I_2 - I_6|\right) \tag{66}$$

3.1.27 Improved binary gradient contours

We propose in this section an extension of BGC1, already described in (Sec. 3.1.23). This improved version includes the central pixel and can be easily derived from the original formulation (Eq. 58) by comparing the central pixel value with the average grey-scale value. The kernel function is:

$$f_{\mathrm{IBGC1}}(\mathbf{x}) = (2^8 - 1)\xi\left(I_c - \bar{S}\right) + \sum_{j=0}^{7} 2^j \xi\left(I_i - I_{(j+1) \bmod 8}\right) - 1 \tag{67}$$

where $\bar{S}$ is the average grey-scale value over the neighbourhood (Eq. 35). We refer to this method as improved binary gradient contours (IBGC1).

3.2 Global methods

3.2.1 Binary texture co-occurrence spectrum

The binary texture co-occurrence spectrum [90] is based on the probability of occurrence of the binary states that a neighbourhood of predefined shape and size (n-tuple) can take once the image is binarized through a global threshold. The implementation adopted here considers the neighbourhood formed by pixels 0, 2, 4 and 6 (Eq. 11). We refer to this method as BTCS+. The kernel function can be expressed as follows:

$$f_{\mathrm{BTCS+}}(\mathbf{x}) = \sum_{j=0}^{3} 2^j \xi \left(I_{2j} - \hat{I}\right) \tag{68}$$

where $\hat{I}$ is the binarization threshold. In our implementation $\hat{I}$ is the value that divides the grey-scale histogram of the whole image in two parts of equal entropy [4].

3.2.2 Coordinated clusters representation

The Coordinated clusters representation (CCR) was originally developed as a texture descriptor for binary images [61]. It was later on extended to grey-scale images through a preliminary thresholding step [99], and, recently, to colour images too [5].

The method is based on the same idea of BTCS, i.e. estimating the probability of occurrence of the 2^9 possible binary instances of a 3×3 window. In some sense it is also similar to LBP and ILBP, the main difference being that threshold is global in this case. The kernel function is:

$$f_{\mathrm{CCR}}(\mathbf{x}) = 2^8 \xi \left(I_c - \hat{I} \right) + \sum_{j=0}^{7} 2^j \xi \left(I_j - \hat{I} \right) \tag{69}$$

where the meaning of $\hat{I}$ is the same as in Eq. 68.

It is interesting to mention that this descriptor has been proposed, apparently independently, by various authors. Chronologically the method should be ascribed to Kurmyshev and Cervantes, who first described it in 1996 [61]. Later the same idea appears in the work of Lindsey and Strömberg [69], where 3×3 binary patterns are referred to as ‘p-grams’. Finally, Kirsanova and Sadovsky [56] described a conceptually equivalent method where binary patterns are referred to as ‘smalts’.

3.2.3 Completed Local Binary Patterns (CLBP_C and CLBP_M)

CLBP_C thresholds the central pixel at the average grey value of the whole image, and therefore generates only two binary patterns. The kernel function is:

$$f_{\mathrm{CLBP\_C}}(\mathbf{x}) = \xi \left(I_c - \bar{I} \right) \tag{70}$$

where

$$\bar{I} = \frac{\sum_{m=1}^{M} \sum_{n=1}^{N} I_{m,n}}{MN} \tag{71}$$

CLBP_M considers the possible binary patterns that are defined by the absolute difference between the gray value of a pixel in the periphery and that of the central pixel when thresholded with a global parameter. In formulas:

$$f_{\mathrm{CLBP\_M}}(\mathbf{x}) = \sum_{j=0}^{7} 2^j \xi \left(|I_j - I_c| - \tilde{I} \right) \tag{72}$$

where $\tilde{I}$ is the average value of the difference in grey value between a pixel in the periphery and the central pixel:

$$\tilde{I} = \frac{\sum_{m=2}^{M-1} \sum_{n=2}^{N-1} \sum_{i=-1}^{1} \sum_{j=-1}^{1} |I_{m-i,n-j} - I_{m,n}|}{8(M-2)(N-2)} \tag{73}$$

3.3 Methods not included in the experiments

Among the methods appeared in literature, there are some that – though belonging in principle to the HEP – could not be included in the experiments due to insufficient description or lack of implementation details. For the sake of completeness, we briefly summarize these methods here below, stating the reasons why we could not include them in the experiments.

The fast coordinated clusters representation (FCCR) [62] is a simplified version of the CCR (Sec. 3.2.2) obtained through row- or column-wise marginalization of the 3×3 neighbourhood. This approach produces 2^3 binary patterns for each row (or column), but the authors do not specify how to combine the histograms corresponding to each row/column.

In Ref. [119] it is described an approach similar to ILBP (Sec. 3.1.12), but with a different procedure to compute the local threshold. The method, referred to as local adaptive Niblack algorithm, computes the threshold as a function of the ratio between the mean value and standard deviation of the grey values in the neighbourhood. The paper does not specify, however, how to treat patterns which have null standard deviation (e.g.: flat areas).

Two variations of a descriptor referred to as fuzzy texture spectrum have been proposed in Refs. [121–123]. Both methods look like extensions of texture spectrum (TSΔ, Sec. 3.1.4) to five and seven quantization levels, respectively. The problem here is that the kernel functions (Eqs. 1 and 2 of Ref. [121]) appear to make use of fractional exponents, which result in irrational numbers. This prevents the method from being inserted in the HEP.

Wu and Sun [125] presented a method based on the joint distribution of the vertical, horizontal and the two diagonal binary triplets obtained by thresholding the corresponding grey-scale values at the average value of the neighbourhood. The mathematical formulation of the method, however – Eq. 3 of Ref. [125] –, is completely unclear and does not match the number of features (105) that the method is supposed to generate.

Finally, the grey level weight matrix (GLWM) [98], based on the ternary thresholding function, is very similar to texture spectrum (TS0, Sec. 3.1.4). The matter with GLWM is that the number of binary patterns that it generates (3^8) does not match the coding scheme described in [98], which can enumerate only 2^8 patterns.

4 Extensions

4.1 Robustness against rotation

The most common procedure to obtain rotationally-invariant descriptors – which in the case of a 3×3 neighbourhood means invariance to discrete rotations of 45° – is based on two steps [85,21]: 1) conversion of the square neighbourhood into a circular one through interpolation, and 2) rotationally-invariant normalization of the feature vector. In the first step the original window (let us assume, as usual, this be a 3×3 window) is replaced by a circular one. Such a neighbourhood is usually indicated as $(8, 1)$, where the first number represents the number of pixels in the periphery and the second the radius (in pixels) of the circle. Grey-scale values of points that do not lie exactly on pixels centres are estimated through bilinear interpolation. In the second step the feature vector is processed to obtain another vector which is invariant against discrete rotations of the input image. To this end two solutions have been proposed: the use of rotationally-invariant patterns and DFT normalization. The first approach groups together the class labels that are equivalent under a circular shift of the index j (see Eq. 11); the second one exploits the property that the DFT of a vector is invariant to any circular shift applied to that vector. For details and technicalities about these two approaches readers are referred to the work of [21]. We recall that the effect of both is dimensionality reduction: in the case of LBP, for instance, the number of features reduces from 2^8 to 36 and 163, when rotationally-invariant patterns or DFT normalization is used.

4.2 Multi-scale analysis

Multi-scale analysis can be theoretically achieved using neighbourhoods wider than the 3×3, for instance: 5×5, 7×7, etc. This approach, however, is impractical, due to the very high-dimensional feature vectors that arise. The usual strategy consists in dividing the neighbourhood into concentric sub-neighbourhoods – though other sampling schemes are also possible [105] – and concatenating the feature vectors of each sub-neighbourhood [85]. Statistically this is a marginalization procedure, which is theoretically correct only under the assumption of independence between the various sub-neighbourhoods. Though this assumption does not hold in general, marginalization has been successfully employed to deal with multi-scale analysis [85]. Multi-scale approaches have been described for GLCM [9,19], LBP [85], and rank-based descriptors [93].

An alternative approach to multi-scale analysis, which is currently quite common, consists in decreasing the image resolution while maintaining the scanning window as small as 3×3. This strategy gives rise to the so-called pyramidal methods. Sun *et al.* [106], for instance, applied CS-LBP to a sequence of increasingly finer spatial grids computed from the original images. Likewise, in the approach proposed by Qian *et al.* [96], a pyramidal descriptor (PLBP) is constructed by cascading the LBP information over a hierarchical spatial pyramid.

4.3 Dimensionality reduction

The feature vector returned by Eq. 2 can be either used as is (this is the approach used in our experiments) or submitted to a post-processing step to further reduce dimensionality. In Sec. 4.1 we have mentioned that some methods to obtain rotationally-invariant descriptors produce, as a side-effect, a reduction in the number of features. In other cases dimensionality reduction can be the sole objective of post-processing. This is very common with methods that return long feature vectors, such as GLCM. In this case the typical treatment consists in extracting global statistical descriptors such as contrast, energy, entropy, correlation, homogeneity, etc. [34,94]. Alternative approaches also exist, such as histogram binarization, which results in a method known as binary co-occurrence matrix (BCM) [59,60]. Post-processing is not limited to GLCM, and has been applied to other methods too, such as texture spectrum [118,39,15] and Local Binary Patterns [68,76].

4.4 Discussion

We conclude this section with some considerations about the possible extensions of the framework. In particular we would like to explain the reasons why we did not considered the extensions in the experiments.

The first reason is that extensions have not been published for all the methods considered in Sec. 3: rotation-invariant and multi-scale variations exist for some methods (e.g.: LBP, ILBP, CCR, etc.), but not for all. Some of the methods considered herein cannot, in fact, be extended to a rotationally-invariant version: consider, for instance, those that employ only one quadrant of the 3 × 3 window (e.g.: STS, MTS). Other methods, such as STU+ and STU×, could be theoretically made rotationally-invariant, but they would actually be invariant to discrete rotations of 90° only.

The second reason is that the use of linear interpolation as a preliminary step to obtain rotationally-invariant descriptors introduces a significant distortion in the space partitioning scheme associated with each method, as set into evidence in Ref. [7]. This distortion is likely to significantly degrade – or at least to alter – the performance of the methods, which is exactly what one seeks to avoid when making comparisons.

Finally, as for post-processing, the problem is that no general approaches exist, but only *ad hoc* solutions tailored to specific descriptors (e.g.: GLCM).

5 Theoretical considerations

In this section we wish to elaborate on two key issues related to texture analysis through 'bag-of-features', a concept introduced in Sec. 2.1. Any method of this type has in fact to answer the following two fundamental questions: 1) whether features should be computed *a priori* (i.e.: independently of any training sample) or *a posteriori* (i.e.: based on a set of training samples); and 2) whether features should come from filter responses or from image patches directly.

5.1 *A priori* vs. *a posteriori*

In the preceding sections we have shown that any method belonging to the HEP is based on a space partitioning function which is defined *a priori*, without the need of any training sample. An alternative strategy, proposed and sustained by various authors, consists in defining partitioning criteria that rely upon a set of representative patterns. In this case the partition is computed *a posteriori*. The definition of the representative patterns is usually referred to as codebook generation [53]. Codebooks are typically created from a set of training images using some standard clustering algorithm such as: k-means [114,115], self-organizing maps [80,113] or vector quantization [83,84]. This procedure generates a class of texture descriptors which Guo *et al.* [32] have referred to as *training-based* (TBM henceforth) methods, to contrast them with *dataset-independent* methods [16], which do not require training images.

These methods can be formally expressed through the same notation adopted in Sec. 2.2. The k-th element of the generic feature vector in the following way:

$$h_k = \frac{1}{D} \sum_{m=m_{min}}^{m_{max}} \sum_{n=n_{min}}^{n_{max}} \delta \left\{ \underset{0 \leq z \leq K-1}{\arg\min} ||\mathbf{x}^{\Omega}_{m,n} - \mathbf{C}_z|| - k \right\} \tag{74}$$

where $\mathbf{C}_z$ represents an entry of the predefined codebook $\mathcal{C}$:

$$\mathcal{C} = \{\mathbf{C}_z \in \mathcal{M}_{3\times3,G} \mid 0 \leq z \leq K-1\} \tag{75}$$

The *a priori*/*a posteriori* dichotomy reflects the ancient philosophical debate as to whether knowledge is attained apart from experience or derives from it. In texture analysis our opinion is that a priori approaches are generally simpler, since they do not require codebook generation. As for a posteriori approaches, these can be very dependent on the training data, with potentially negative consequence on classification (e.g.: overfitting). Moreover, care should be taken in choosing the appropriate clustering approach to generate codebooks: the use of non-deterministic procedures (e.g.: k-means) may lead to non-repeatable and therefore unpredictable results.

5.2 Image patches vs. filter responses

The second dichotomy deals with the kind of data that descriptors take as input. These can be either filter responses or image patches. Filter responses have been employed since the dawn of texture analysis. Their use has been justified in various ways. Unser [111] affirmed that, theoretically, any transform of the original image patches satisfying some optimality criteria should always improve the efficiency, when compared with the original representation. Griffin and Lillholm [29] established a relationship between the quantitative description provided by a family of linear filters and the qualitative description of local image structure. Furthermore, certain classes of filters (e.g. Gabor filters) are considered to be closely related to the human vision system [17].

Other authors, however, have set into evidence that texture classification can also be approached, simply and effectively, by employing image patch exemplars with a support as small as 3 × 3, without the use of large filter banks [114]. In a recent work Ghita *et al.* [25] have shown that the performances in texture classification offered by LBP/C and multi-channel Gabor filtering are comparable.

In principle there are advantages to both strategies, and, to the best of our knowledge, none of the two has been definitely proven superior. The use of image patches certainly reduces the computational overhead and frees the user from having to design complex filter banks that are likely to be application-dependent [70]. Conversely, methods based on filtering tend to be more robust to noise, since the change in the value of a single pixel – which can dramatically affect patch-based descriptors [132] – has less effect in this case. Moreover, specific filters such as Local Phase Quantization [87,43] can be used to implement blur-invariant texture descriptors.

6 Experiments

To compare the performance of the texture descriptors presented in Sec. 3 we submitted them to an extensive texture classification experiment. In the following subsections we describe the datasets considered in the experiments and the methodology through which the performance of the methods has been assessed.

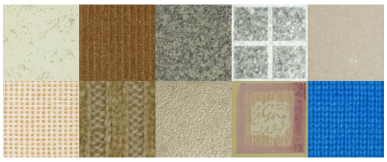
6.1 Datasets

Eleven image datasets have been considered in this experiment. The main characteristics of each dataset are described in detail in the following subsections and summarized in Tab. 2.

6.1.1 Bonn BTF

The BTF Database Bonn [8,36], maintained at the Institute of Computer Science II of the University of Bonn (Germany), is a part of a project aiming to develop novel techniques for the efficient and high-fidelity capture of materials' appearance. The image acquisition process is based on an impressive multi-camera apparatus consisting of 151 digital consumer cameras mounted on a dome-shaped rack. This enables simultaneous acquisition under varying viewing and illumination angle. The dataset considered here (Fig. 1) is obtained by merging two different datasets which are part of BTF Database Bonn: 'ATRIUM' and 'UBO2003'. The former contains four texture classes of floors and coverings: ceiling, walkway, floortile and pinktile; the latter six classes of general materials: corduroy, granite, upholstery, polyacryl, wallpaper and wool. In both cases the images considered herein are those taken at orthogonal viewing and illumination angle. The ATRIUM images have been cropped to 256 × 256 first, and then subdivided into 16 non-overlapping sub-images. The UBO2003 images, whose original resolution is 800 × 800, have been subdivided into 16 non-overlapping sub-images. Though this dataset includes images of different resolution, this does not affect the experiments, since all the texture descriptors provide normalized histograms (see Eq. 2).

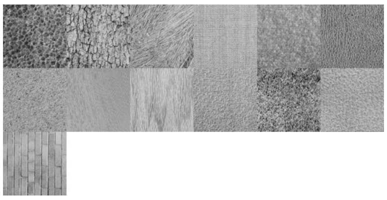
Fig. 1: The 10 classes of the Bonn BTF dataset.



6.1.2 Brodatz

Brodatz is one of the first datasets that have been used in texture analysis. It includes textures of natural scenes and materials (i.e.: grass, bark, sand and straw) as well as artificial manufactured goods (i.e.: raffia, pigskin and bricks). Despite it is quite old, it is largely used still today. It is important to point out that the original source images [10] are not in digital format. Various digital repositories exist, though they are not exactly equivalent. When using this dataset it is therefore recommendable to specify the exact source. The images used here have been downloaded from a repository [112] maintained by the University of Southern California. The dataset (Fig. 2) includes the following classes (one image per class): D9, D12, D15 D16, D19, D24, D29, D38, D68, D84, D92, D94 and D112. In order to obtain more samples for each class, the original images, the resolution of which is 1024 × 1024, have been subdivided into 16 non-overlapping sub-images of resolution 256 × 256.

Fig. 2: The 13 classes of the Brodatz dataset.



6.1.3 Jerry Wu

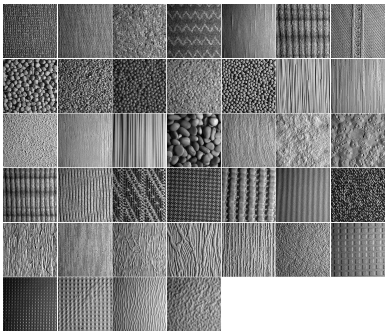
This database has been developed within the TextureLab at the Heriot-Watt University (UK). The name

Table 2: Summary of the datasets considered in the experiments.

Name	Number	Classes	Samples per class	Total samples	Sample resolution (pixels)	Image format	Predefined train/test sets?
Bonn BTF	1	10	16	160	200 × 200 and 64 × 64	Colour (JPEG)	No
Brodatz	2	13	16	208	256 × 256	Monochrome (TIFF)	No
Jerry Wu	3	39	4	156	256 × 256	Colour (BMP)	No
KTH-TIPS	4	10	4	40	100 × 100	Monochrome (PNG)	No
KTH-TIPS2b	5	11	16	176	100 × 100	Colour (PNG)	No
MondialMarmi	6	12	64	768	136 × 136	Colour (BMP)	No
OuTeX TC_00000	7	24	20	480	128 × 128	Monochrome (RAS)	Yes
OuTeX TC_00001	8	24	88	2112	64 × 64	Monochrome (RAS)	Yes
OuTeX TC_00013	9	68	20	1360	128 × 128	Colour (BMP)	No
UIUCTex	10	25	40	1000	640 × 480	Monochrome (JPEG)	No
VisTex	11	167	16	2672	128 × 128	Colour (BMP)	No

comes from Jerry Wu, the researcher who built it as a part of his PhD [124]. The dataset includes 39 texture classes acquired under different combinations of illumination direction, imaging direction and surface rotation [49]. They include natural and artificial materials such as wood, wallpaper, fabric and grains. The original images are grey-scale with a resolution of 512 × 512. The subset used in our experiments (Fig. 3) contains the images captured under slant angle 0°, tilt angle 45° and surface orientation 0°. Each original image has been subdivided into four non-overlapping sub-images.

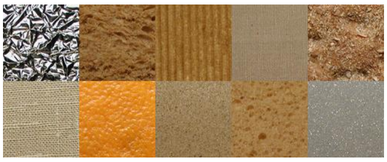
Fig. 3: The 39 classes of the Jerry Wu dataset.



6.1.4 KTH-TIPS

The KTH-TIPS database [57] was designed within the Computer Vision and Active Perception Lab of the KTH Royal Institute of Technology (Sweden) as a tool to investigate the effect of real-world imaging conditions on material classification [37]. The dataset includes 10 classes of common materials: aluminium foil, bread, corduroy, cotton, cracker, linen, orange peel, sandpaper, sponge and styrofoam. Images are taken under nine equally-spaced scales over two octaves, three rotation angles and three lighting directions, giving 81 images for each class. The subset considered here includes the images acquired at scale 5 (the central scale, corresponding to an object/camera distance of 28.0 cm), illumination and pose angle 0°. The resolution is 200 × 200. A further division into four 100 × 100 sub-images has been performed to get more samples per class.

Fig. 4: The 10 classes of the KTH-TIPS dataset.



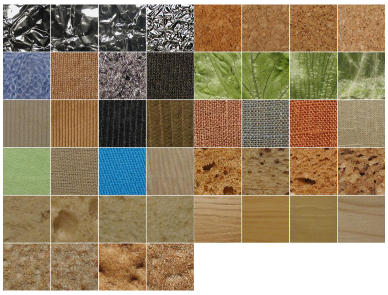
6.1.5 KTH-TIPS2b

The KTH-TIPS2b database [11] is an extension of KTH-TIPS. The acquisition process largely follows the procedure used for KTH-TIPS, with some differences regarding scale and illuminant. KTH-TIPS2b presents 11 classes of materials with 16 samples per class (Fig. 5). The subset used here maintains the same settings (i.e.: object/camera distance, illumination and pose angle used for KTH-TIPS – see Sec. 6.1.4).

6.1.6 MondialMarmi

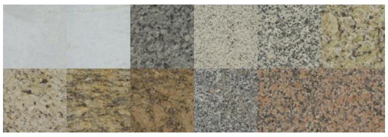
MondialMarmi is a database of images of granite tiles (Fig. 6) for colour and texture analysis. The current version (1.1) features 12 granite classes: Acquamarina, Azul Capixaba, Azul Platino, Bianco Cristal, Bianco Sardo, Giallo Napoletano, Giallo Ornamentale, Giallo Santa Cecilia, Giallo Veneziano, Rosa Beta, Rosa Porriño A, Rosa Porriño B. There are four images for each class. The images, available as colour .bmp (resolution:

Fig. 5: The 11 classes of the KTH-TIPS2b dataset.



544 × 544), have been acquired under controlled illumination conditions using a dome illuminator and a consumer digital camera. The acquisition procedure is described in detail in Ref. [21]. Scale and illumination are invariable, but the dataset includes hardware- and software-rotated images. The dataset can be accessed freely at the URL indicated in Ref. [72].

Fig. 6: The 12 classes of the MondialMarmi dataset.



6.1.7 OuTeX TC_00000 and OuTeX TC_00001

Outex is a well-known general framework for evaluating texture classification and segmentation algorithms [86,88]. The image dataset contains a wide variety of surface textures acquired under controlled and variable conditions of illumination, rotation and spatial resolution. The framework also provides pre-defined classification and segmentation problems (test suites). OuTeX TC_00000 and OuTeX TC_00001 present the same 24 texture classes (Fig. 7), but the number of samples for each class is different (20 and 88, respectively). The image resolution is: 128 × 128 for OuTeX TC_00000 and 64 × 64 for OuTeX TC_00001. In both datasets the imaging conditions (i.e.: type of illuminant, rotation and scale) are invariable.

6.1.8 OuTeX TC_00013

The OuTeX TC_00013 test suite (Fig. 8) contains 68 colour texture images acquired under invariable illumination, rotation and scale. Image resolution is 128 × 128 pixels. To obtain more samples for each class the original images have been subdivided into four non-overlapping sub-images.

Fig. 7: The 24 classes of the OuTeX TC_00000 and OuTeX TC_00001 test suites.

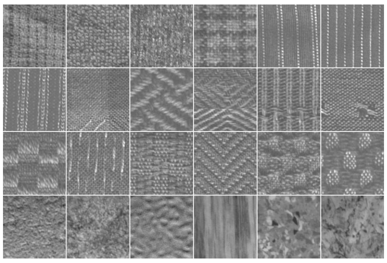
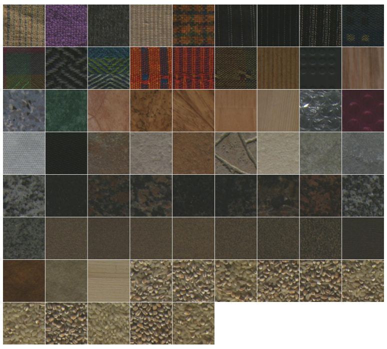


Fig. 8: The 68 classes of the OuTeX TC_00013 test suite.



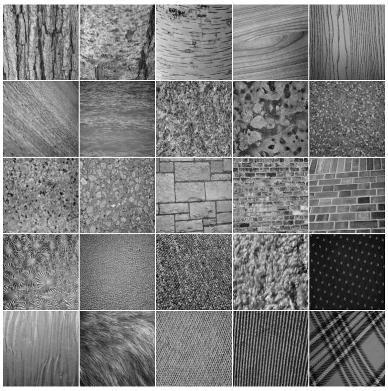
6.1.9 UIUCTex

The UIUCTex texture database [64] (Fig. 9), developed and maintained by the Ponce Group at the University of Illinois at Urbana-Champaign, features 25 texture classes of natural and artificial materials such as bark, wood, glass, marble, fabric, etc. For each class the dataset contains 40 samples acquired under variable – but uncontrolled – imaging conditions. The images are grey-scale with a resolution of 640 × 480 pixels and can be downloaded at the URL reported in Ref. [109]. This dataset is particularly challenging since it includes affine transforms, viewpoint and illumination changes and non-rigid deformations.

6.1.10 VisTex

The Vision Texture database (VisTex) is a collection of texture images developed at the Massachusetts Institute of Technology (USA). Since Dec. 2002 the database is no longer maintained, but only available ‘as is’ [117].

Fig. 9: The 25 classes of the UIUCTex dataset.



The dataset (Fig. 10) contains 167 texture classes – one image per class – representing quite heterogeneous materials and scenes, such as grass, water, brick, buildings, clouds, sand, etc. The images have been acquired under uncontrolled illumination and viewing conditions [102]. Each image, whose original resolution is 512 × 512, has been subdivided into 16 non-overlapping sub-images.

Fig. 10: The 167 classes of the VisTex dataset.

6.2 Comparative assessment of performance

A comparative analysis of the texture descriptors presented in Sec. 3 and summarized in Tab. 1 has been carried out through a supervised image classification task. Assessing the relative performance of texture descriptors is not straightforward in this case, since the methods are evaluated over several datasets. A mere comparison of the average classification accuracy would make little sense here, due the diversity of the datasets, and would possibly lead to inconclusive results. We therefore adopted a procedure in three steps: 1) estimation of the accuracy of each descriptor for each dataset; 2) pairwise comparison and 3) ranking.

In the first step the images of each dataset are preliminarily divided into two disjoint sub-sets, one for training and the other for validation, with the constraint that one half of the samples of each class is used for training and the other half for validation. The images of the validation set are then classified through the nearest-neighbour rule (1-NN) with L_1 distance. The estimated accuracy is the percentage of images of the validation set which are classified correctly. The procedure is repeated 100 times, each time with a different subdivision into training and validation set, and the accuracy obtained with each subdivision is recorded. The same 100 subdivisions into training and validation set are used for all the texture descriptors. As for OuTeX TC_00000 and OuTeX TC_00001, we maintained the same subdivisions the two test suites come with.

The second step deals with the relative comparison of pairs of methods within each dataset. Let $\mathbf{a}_1$ and $\mathbf{a}_2$ be the accuracy vectors (100 elements each) of the two descriptors that are under comparison. To check whether there is significant difference between the two methods we used the Wilcoxon signed-rank test [73] with $\alpha = 0.05$. The positiveness of this non-parametric test rejects the hypothesis that the components of the vector $(\mathbf{a}_1 - \mathbf{a}_2)$ are randomly drawn from a symmetric, continuous and null-median distribution. Therefore if the test detects significant difference we consider that one of the two methods outperforms (or underperforms) the other.

In the third step we construct a ranking based on the following rule: for each pairwise comparison, if one of the two methods outperforms the other, assign a +1 to the winner. Finally the points that a method obtains in each dataset are summed up to provide the overall ranking.

We conclude this section with some considerations in support of the use of the nearest-neighbour classifier. As it has been set into evidence in a recent work [6], this classification strategy is particularly suitable for feature comparison purposes due to the absence of tuning parameters, easiness of implementation and other desirable asymptotic properties. A review of recent related literature indeed shows that 1-NN is most commonly adopted in evaluating the relative performance of texture analysis algorithms [115, 16, 31, 54, 70].

More sophisticated methods may provide better absolute results, but at the cost of complicated tuning procedures. For comparative purposes we repeated the experiments using SVM, a classifier which is usually

Table 3: Average classification accuracy.

Texture descriptor	Dataset										
	1	2	3	4	5	6	7	8	9	10	11
3DLBP	88.42	99.21	95.19	99.75	88.99	85.53	96.38	97.25	76.71	58.90	73.34
BGC1	96.66	100.00	97.59	100.00	90.33	82.67	99.90	98.54	79.09	58.66	76.19
BGC2	96.45	99.39	97.09	100.00	82.82	75.41	99.13	96.68	76.81	52.37	69.87
BGC3	96.91	99.80	98.21	99.10	84.64	76.93	98.83	96.32	75.93	52.47	68.51
BTCS+	93.11	96.23	84.58	98.55	78.06	72.01	89.65	84.05	63.21	41.21	41.16
CBP	97.64	99.19	91.04	99.05	87.27	84.28	97.26	95.09	72.43	57.99	61.87
CCR	92.45	98.76	93.71	100.00	84.43	80.88	96.30	91.91	70.73	53.07	53.37
CLBP_M	84.58	98.95	95.06	100.00	76.95	82.30	98.43	96.45	71.72	54.48	65.21
CLBP_C × CLBP_M	87.63	99.88	97.37	99.55	87.60	87.39	99.10	98.74	78.12	68.82	67.45
CLBP_S || (CLBP_M × CLBP_C)	92.89	100.00	98.71	100.00	94.07	92.35	99.84	99.51	81.94	73.58	75.85
CLBP_M × CLBP_S	59.51	100.00	99.59	99.60	91.84	73.53	99.88	96.89	76.93	76.51	75.65
CLBP_C × CLBP_M × CLBP_S	68.44	100.00	99.60	98.25	91.03	76.74	99.94	97.07	78.47	82.29	74.68
CS-LBP	92.63	99.01	91.71	100.00	81.16	75.09	92.95	86.77	64.88	44.39	54.26
CS-TSΔ	97.55	99.67	95.42	100.00	88.78	84.08	98.05	95.95	73.64	57.02	65.95
D-LBP	99.58	99.93	92.94	97.30	82.65	78.29	97.02	90.81	73.09	45.47	66.15
GLBP	83.01	99.86	95.32	99.40	80.50	74.44	97.27	92.80	71.15	45.46	68.40
GLCM	99.55	96.98	68.92	92.55	83.94	88.69	38.49	43.62	79.88	45.98	77.41
GLD	80.40	79.99	59.01	84.15	75.26	60.60	56.52	65.57	45.47	37.33	54.91
GLTCS+	95.47	97.98	93.10	96.70	72.24	66.88	97.22	90.97	71.59	44.23	58.34
GTUC	94.79	98.81	92.22	98.75	86.83	82.57	97.99	96.51	71.79	54.20	65.81
IBGC1	95.90	100.00	98.27	100.00	93.10	85.34	99.95	99.23	80.19	61.16	77.88
ICS-TSΔ	94.81	97.03	90.82	93.80	84.73	77.92	95.53	91.91	67.46	44.10	50.71
ID-LBP	97.59	97.18	88.51	99.65	75.45	73.25	95.90	88.50	68.85	37.33	57.71
ILBP	95.76	100.00	98.64	100.00	92.08	86.63	99.61	99.36	80.31	61.62	75.41
ILTP	99.17	100.00	98.85	100.00	93.93	91.19	99.87	99.58	80.88	66.84	77.89
LBP	95.86	100.00	97.26	100.00	89.68	83.42	99.68	98.43	78.18	57.25	74.20
LQP	98.17	99.99	98.27	100.00	93.17	87.82	99.13	98.92	79.36	66.31	76.42
LTP	99.00	100.00	98.45	100.00	93.80	88.84	99.75	99.22	80.05	63.96	76.46
MBP	98.61	99.97	96.65	100.00	88.91	85.46	99.35	97.67	77.73	58.54	74.29
MTS	92.08	97.48	91.71	91.30	75.30	69.24	97.52	91.35	69.88	44.30	56.64
RT	94.26	99.97	91.13	100.00	77.10	67.06	94.25	84.36	69.17	57.05	54.11
RTU	97.45	99.79	94.40	97.65	82.27	74.58	94.72	89.59	72.77	62.99	64.77
SDH	99.56	98.04	72.65	94.00	87.33	91.32	58.86	67.03	83.21	69.35	71.11
STS	93.30	99.33	94.82	99.15	82.33	74.07	98.29	95.49	74.53	49.26	66.34
STU+	99.81	99.97	97.06	100.00	91.93	87.27	98.87	98.32	77.47	62.11	73.35
STU×	98.35	99.99	98.73	100.00	92.20	89.86	99.02	98.81	79.70	65.82	72.94
TS0	96.58	100.00	97.71	100.00	89.80	85.22	99.48	97.79	79.32	63.02	76.49
TSΔ	95.16	100.00	98.95	100.00	93.82	89.66	99.66	98.94	80.39	68.66	78.78
BIF-M	88.63	99.85	97.53	98.25	75.31	60.62	97.88	91.75	69.35	94.39	55.78
BIF-W (L_2)	94.91	98.33	93.41	93.55	67.16	51.09	95.70	84.87	59.83	85.93	45.41

considered superior to the 1-NN. The problem is that the accuracy of an SVM model is largely dependent on the selection of the model's parameters, particularly C and γ [14]. As for texture classification problems, different values of C and γ have indeed been reported in literature. Li *et al.* [67] proposed the following pairs of values: ($C = 1000$, $\gamma = 50$), ($C = 1000$, $\gamma = 0.5$) and ($C = 1000$, $\gamma = 0.02$); Kim *et al.* [55] adopted ($C = 100$, $\gamma = 2$), and Rajpoot and Rajpoot [97] ($C = 1$, $\gamma = 0.001$). Our experiments conducted with SVM using radial-basis kernel and these five couples of values showed that classification accuracy is very sensitive to these parameters, and that performance drops drastically when the SVM model is not tuned properly. To give an example consider that the accuracy of ILTP (best performer) on the KTH-TIPS2b dataset drops from 93.50% to 56.44%, while the accuracy of LTP (second-best performer) drops from 90.34% to 54.19% when SVM parameters switch from ($C = 1000$, $\gamma = 50$) to ($C = 1$, $\gamma = 0.001$). The SVM classification results about the first 10 methods of the ranking can be accessed here [44]. We notice that not only the accuracy changes significantly, but that the relative ranking changes as well. These findings strongly support, in our opinion, the use of a parameter-free classifier like the 1-NN for comparative purposes.

6.3 Methods that depend on parameters

Some of the methods considered in the experiments depend on one or two parameters which are used to feed the ternary (Eq. 9) or quinary (Eq. 10) thresholding functions. In the text these parameters are indicated as Δ, Δ_1 and Δ_2, respectively. Since the values of these user-specified thresholds have significant effects on the performance, particular attention should be taken in evaluating these methods. The problem is that no general criterion to estimate optimal values for these parameters has been proposed so far. The values available in literature are indeed quite different. Madrid-Cuevas *et al.* [71] report optimal results for TS$\varDelta$, STU+ and STU× with $\varDelta = 16$, 8 and 32, respectively, on an image classification task based on the Brodatz album. Tan and Triggs [107] suggested using $\varDelta = 5$ for face recognition based on LTP. Chang and Chen [13] used $\varDelta = 3$ for a texture classification experiment based on TS$\varDelta$ and GTUC. Finally, some authors [40,24] do not specify the value of $\varDelta$ at all, but simply refer to it as a "small positive value". Even more complicated is the case when two parameters are involved, as in local quinary patterns. In the implementation presented in Ref. [74] the authors proposed $\varDelta_1 = 2$ and $\varDelta_2 = 5$.

The procedure adopted herein is based on an exhaustive search over a predefined set of parameter values. As a first step we established a predefined set of values for the parameters: $\varDelta \in [1, 32]$, $\varDelta_1 \in [1, 3]$ and $\varDelta_2 \in [4, 6]$. Then we estimated the accuracy of each method with each dataset for each value (or couple of values) of the parameters with the procedure described in the preceding section. Finally, for each descriptor and dataset we picked out best results, which are those reported in Tab. 3. The parameters that yield the best accuracy are reported in Tab. 4 (in some cases the parameters that give optimal results are not unique).

6.4 Implementation, execution and reproducible research

The texture descriptors have been implemented in MATLAB® R2008b. The classification experiments have been performed in the School of Industrial Engineering, at the University of Vigo, on a PC equipped with INTEL® CORE™ Quad CPU Q8200, 4GB RAM, and WINDOWS™ 7 – 64 bits, Service Pack 1. The execution of the scripts required more than 350 hours of computing time. For reproducible research purposes, all the data required to replicate the experiments (i.e.: source code, images and subdivisions into train and validation sets) are available in Ref. [44][1].

6.5 Results

The experimental results are summarized in Tab. 3 and Fig. 11. The former reports the average accuracy (i.e.: over the 100 problems) of each method and for each texture dataset. These results give an idea of the overall performance of the descriptors in the various datasets. For comparative purposes, however, it is recommendable to consider the final ranking, as explained in Sec. 6.2. This is reported in the form of a scatter plot (Fig. 11) in which the x-axis represents the dimension of the feature vector (in $\log_2$ scale) and the y-axis the normalized number of victories obtained by each method (see Sec. 6.2 for details). The tournament shows that ILTP and CLBP_S || (CLBP_M × CLBP_C) are the best among the parametric and non-parametric methods, respectively. The results also show the following patterns:

- Multilevel discretization is more effective than pure binarization. Seven out of the first ten positions of the ranking are indeed occupied by local methods based on ternary or higher-order thresholding schemes, such as ILTP (1^{th}), LTP (2^{nd}), TS$\varDelta$ (4^{th}), LQP (5^{th}), STU× (7^{th}), TS0 (9^{th}) and STU+ (10^{th}).
- Parametric descriptors are more accurate than non-parametric ones. A representative example of this trend is the better performance of TS$\varDelta$ (parametric) with respect to its non-parametric counterpart (TS0). In evaluating the performances of parametric methods we should not forget, however, that we picked out the optimal results over a predefined set of parameter values (Sec. 6.3). These results must be therefore considered 'optimized', since the parameter values are tuned for each dataset.
- There is a general trend of increasing performance with increasing dimensionality (Fig. 11). Descriptors that considerably deviate from this trend are STU+ and STU× on one side (they perform remarkably well with few features), and GLCM on the other side (they perform rather poorly with many features).
- Methods based on point-to-average thresholding outperform those based on point-to-point thresholding. Fig. 11 clearly shows that descriptors which use the grey level of a single pixel as a threshold (such as LTP, BGC1 and LBP) are less accurate than their 'improved' counterparts (namely ILTP, IBGC1 and ILBP), in which the threshold is computed from the entire neighbourhood. This trend is likely to be related to the thresholding method itself, though the associated increase in dimensionality could as well contribute to the improvement.

The experimental activity presented in this section is mainly intended to establish a relative ranking of texture descriptors within the HEP. Still, this leaves the following question unsolved: how good are methods belonging to the HEP when compared with the wider universe of texture descriptors? To address this issue we calibrated the performance of the HEP against basic image features (BIF-columns) [16]. As we mentioned in Sec. 2.2, BIF is conceptually related to the HEP – both are based on space partitioning – and therefore

[1] To access the page: user = texture, psw = analysis

Table 4: Optimal parameter values.

Texture descriptor	Parameters	Dataset										
		1	2	3	4	5	6	7	8	9	10	11
CBP	$\varDelta$	4	16	7	$\{12, 14, 15\}$	12	11	8	13	4	12	8
CS-LBP	$\varDelta$	6	4	5	[6, 8]	9	17	2	11	3	4	6
CS-TS$\varDelta$	$\varDelta$	5	3	9	5	9	8	10	10	4	12	7
GTUC	$\varDelta$	8	13	6	4	10	8	6	11	4	9	7
ICS-TS$\varDelta$	$\varDelta$	3	10	6	8	8	8	7	11	3	9	9
ILTP	$\varDelta$	2	[1,3]	1	[1,7]	2	2	$\{1, 2\}$	1	1	2	1
LQP	$\varDelta_1$,$\varDelta_2$	3,4	3,4	3,4	3,{4,6}	2,6	1,6	2,6	1,5	1,5	3,5	3,4
LTP	$\varDelta$	3	[1,7]	5	[1,15]	7	7	2	5	2	6	2
STU+	$\varDelta$	2	3	1	$\{1, 3, 6, 7\}$	10	4	4	4	2	4	2
STUx	$\varDelta$	1	3	3	$[1,8] \cup 11$	4	5	4	7	1	5	3
TS$\varDelta$	$\varDelta$	7	[1,3]	5	[2,12]	4	2	1	4	1	4	2

the comparison is meaningful. Moreover, BIF showed the best performance over three datasets at the time Ref. [16] was submitted. Yet, there are considerable differences that one should not forget: BIF is a multi-scale (here we used four octave-spaced scales, as in [16]) and filtering-based method, whereas the methods belonging to the HEP are single-scale (3 × 3) and and employ no filtering. As a consequence, basic image features need a significantly larger support to be computed, a requirement that can negatively affect the results when the images are small. This inconvenience can be solved either by mirroring the image when filters overhang the borders, or through toroidal wrapping. In the experiments we employed the default implementation, which is based on image mirroring, as provided to the authors by the research group where BIF where developed. We refer to this method as BIF-M. In order to make a fair comparison, we maintained the same metric used for the HEP (i.e. L_1 distance). For the sake of completeness, we also considered the implementation based on toroidal wrapping, which is the approach used in Ref. [16]). In this case similarity between BIF-columns is evaluated through the Euclidean distance, as advocated in Ref. [30]. We refer to this method as BIF-W (L_2). Our calibration reveals that, on average, the comparison HEP vs. BIF leans in favour of the former (Fig. 11). Remarkable exception is the result obtained with the UIUCTex dataset (Sec. 6.1.9): here we notice that BIF clearly outperforms the other methods (Tab. 3). As we mentioned above, this peculiar result is likely to be related to image size: of the datasets considered here UIUCTex is in fact the one with the largest image samples.

We conclude this section with a final consideration. In our view, most of the research related to texture analysis has followed, up to now, a merely heuristic approach, i.e.: a texture descriptor is considered ‘good’ or ‘bad’ on the basis of the experimental accuracy alone, without any further investigation about its theoretical foundations. There are, though, some remarkable exceptions, such as CLBP [31], which the authors justify in terms on certain signal reconstruction properties, BGC [22], which is shown to implement a high-entropy space partitioning strategy and CCR [61] – though in this case the analysis is limited to the reconstruction properties of the method with binary and periodic texture. The fact that the first two methods appear in the first 11 positions of the ranking is not accidental, in our view, suggesting that experimental accuracy and firm theoretical background are closely related.

6.6 Effects of resolution and type of patterns

In Sec. 4.4 we have mentioned that rotation-invariant and multi-scale variations have been proposed only for a small subset of the methods presented in Sec 3. It is for this reason that we limited our comparison to the basic 3 × 3 versions. Multi-resolution and rotationally-invariant formulations, however, are likely to affect the results. LBP is perhaps the method that has been studied most extensively, under this perspective, and that triggered the highest number of variations [65]. For the sake of completeness we compared the results of LBP variants on some of the datasets to show the influence of multi-resolution and of the different kind of patterns on the performance of the method (Tab. 5). LBP variants are indicated as in Ref. [85].

The results show that resolution and pattern type have an appreciable effect on the performance. We see that in general $\mathrm{LBP}_{3\times3}$ outperforms the other variants. Indeed, if we extend the experiment to all datasets (the complete results have been omitted to avoid unnecessary burden), and compute the ranking of LBP variants, we find that the first three positions are occupied, respectively, by: $\mathrm{LBP}_{3\times3}$ (1^{st}), $\mathrm{LBP}^{ri}_{8,1}$ $||$ $\mathrm{LBP}^{riu2}_{16,2}$ $||$ $\mathrm{LBP}^{riu2}_{24,3}$ (2^{nd}) and $\mathrm{LBP}^{ri}_{8,1}$ $||$ $\mathrm{LBP}^{riu2}_{16,2}$ (3^{rd}). The fact that LBP3 × 3 leads the ranking of LBP variants rein-

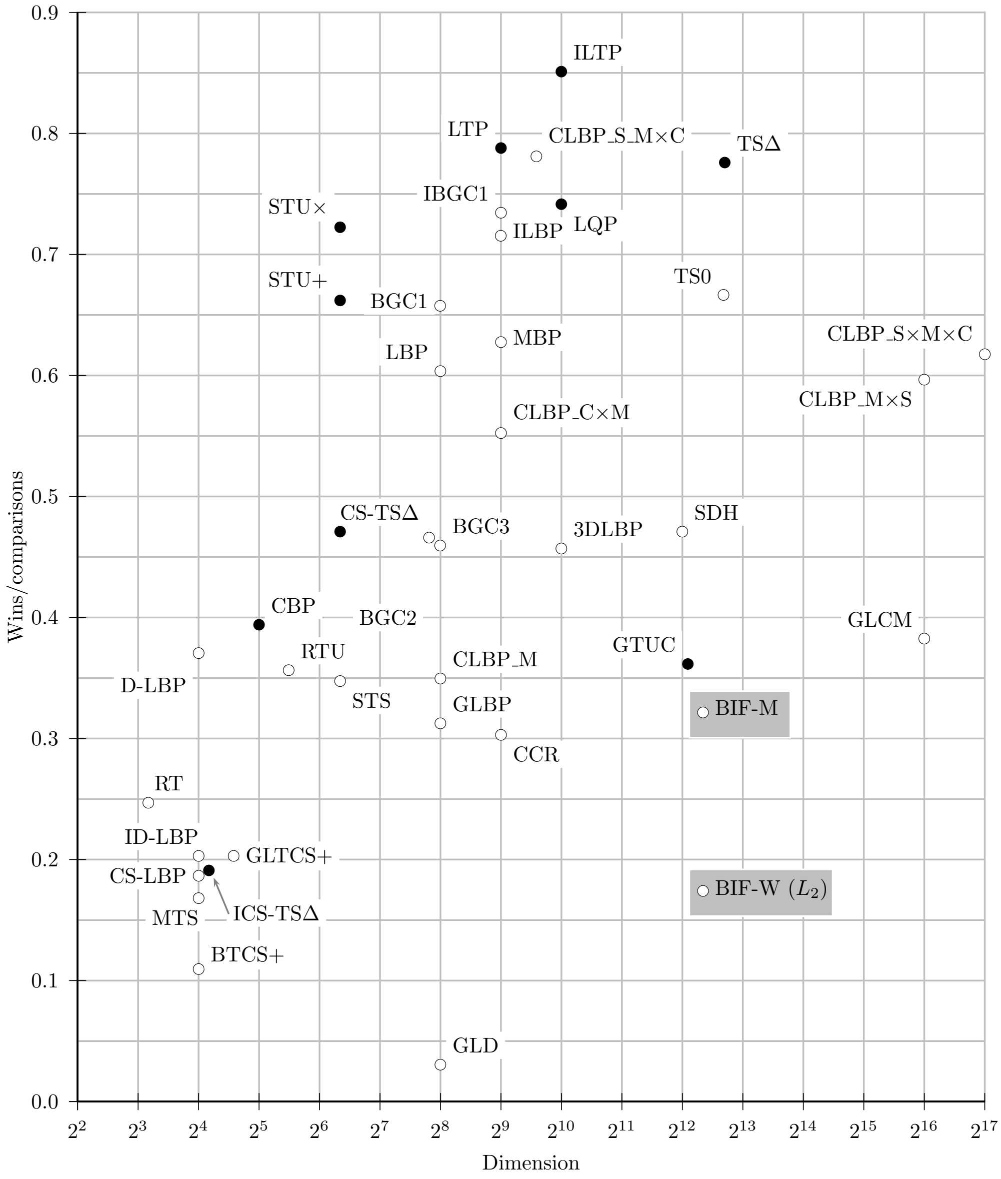


Fig. 11: Ranking. Abbreviations: CLBP_S_M×C = CLBP_S || (CLBP_M × CLBP_C); CLBP_S×M×C = CLBP_S × CLBP_M × CLBP_C; CLBP_C×M = CLBP_C × CLBP_M; CLBP_M×S = CLBP_M × CLBP_S. Black dots indicate parametric methods.

Table 5: Performance of LBP variants.

LBP variant	Dataset			
	3	5	6	9
$\mathrm{LBP}_{3\times3}$	97.26	89.67	83.51	78.22
$\mathrm{LBP}^{ri}_{8,1}$	91.24	78.81	69.83	72.42
$\mathrm{LBP}^{ri}_{16,2}$	98.04	82.13	68.15	74.34
$\mathrm{LBP}^{riu2}_{8,1}$	87.53	78.88	69.32	70.79
$\mathrm{LBP}^{riu2}_{16,2}$	96.29	80.81	67.60	73.56
$\mathrm{LBP}^{riu2}_{24,3}$	95.36	77.13	69.68	73.23
$\mathrm{LBP}^{riu2}_{8,1} \parallel \mathrm{LBP}^{riu2}_{16,2}$	96.28	84.30	79.50	77.77
$\mathrm{LBP}^{riu2}_{8,1} \parallel \mathrm{LBP}^{riu2}_{16,2} \parallel \mathrm{LBP}^{riu2}_{24,3}$	97.59	86.03	82.06	79.58

forces the use of the 3×3 implementation as a basis for the comparison presented in Sec. 6.5.

7 Conclusions and perspectives

In this paper we have described a general framework for texture analysis which we refer to as HEP (Histograms of Equivalent Patterns). The common trait of the methods belonging to the HEP is that each texture descriptor determines a partition of the grey-scale pattern space into equivalence classes. Such a partition is established a priori in the HEP, and is based on the definition of a local or global function (kernel function) of the pixel values. This approach is therefore alternative to those based on a posteriori space partitioning, which is typical of methods based on codebooks learnt from data. Diverse texture descriptors can be seen all to be examples of HEP. We have indeed revisited an ample set of apparently diverse methods and showed that they can be represented in a very neat manner within the HEP. In the experimental part we have carried out a performance characterization of all such published schemes over a wide range of datasets and identified the best descriptors as ILTP among the parametric methods and CLBP_S || (CLBP_M × CLBP_C) among the non-parametric ones. We also observed the following general patterns: 1) multi-level discretization is more effective than binarization; 2) the higher accuracy of parametric methods when compared to non-parametric ones; 3) a general trend of increasing performance with increasing dimensionality; and 4) methods based on point-to-average thresholding outperform their counterparts based on point-to-point thresholding – a trend partly explainable by the associated increase in dimensionality. With regard to the second point, however, is it fair to mention that parametric methods entail the disadvantage of needing to establish, beforehand, optimal values for the Δs. In practical applications their use is recommendable when one has at his own disposal enough data to preliminary tune the parameter's values. Calibration against state-of-the-art basic image features shows that the best HEP methods are better than BIF-columns for small images, but this trend is reversed for larger images (see Tab. 3).

We would like to emphasize once more that the establishment of a texture descriptor belonging to the HEP is a matter of defining an appropriate kernel function which induces a partition of the grey-scale pattern space into equivalence classes. We believe that this idea opens up appealing perspectives for future research. A possible approach could be the study of functions with some desirable properties, such as invariance against rotation, grey-scale changes or other transformations, and maximization of the amount of information (entropy) associated with the partitioning scheme. Another direction for future investigation is related to parameter optimization for those methods (e.g.: TSΔ, LTP, LQP, etc.) that depend on one or more parameters. At present the values of these parameters are in fact determined through trial-and-error, and no criterion is available to compute them on a theoretical basis.

Finally, it is fair to mention that in the experiments we have used several dataset containing rather similar kinds of images. In most cases these represent natural/artificial materials and natural scenes. Therefore the above conclusions may not directly extend to applications such as texture-based satellite imagery segmentation or pedestrian detection. A direction for future work could be the extension of the present study to these domains.

Acknowledgements This work was partially supported by Ministero dell'Istruzione, dell'Università e della Ricerca (Italy) and Mondial Marmi S.r.l. (Italy) within the research project no. 39554 entitled *Expert system for automatic visual inspection of natural stone products* and by the Spanish Government under projects no. TRA2011-29454-C03-01 and CTM2010-16573. The authors wish to thank Prof. Lewis Griffin of University College London for providing them with the Matlab® implementation of basic image features.

References

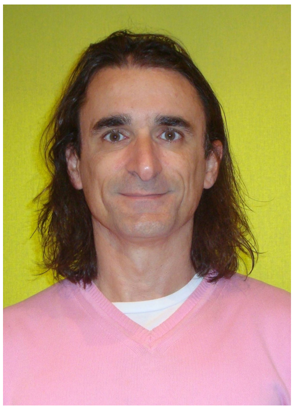
1. Arndt, J.: Matters Computational: Ideas, Algorithms, Source Code. Springer, Berlin (2010)
2. Austin, J.: Grey scale N tuple processing. In: J. Kittler (ed.) Pattern Recognition: 4th International Conference, *Lecture Notes in Computer Science*, vol. 301, pp. 110–120. Springer-Verlag, Cambridge, UK (1988)
3. Beck, M., Robins, S.: Computing the Continuous Discretely. Integer-point Enumeration in Polyhedra. Springer, New York (2007)
4. Bianconi, F., Fernández, A., González, E., Ribas, F.: Texture classification through combination of sequential colour texture classifiers. In: L. Rueda, D. Mery, J. Kittler (eds.) Progress in Pattern Recognition, Image Analysis and Applications. Proceedings of the 12th Iberoamerican Congress on Pattern Recognition (CIARP 2007), *Lecture Notes in Computer Science*, vol. 4756, pp. 231–240. Springer (2008). DOI 10.1007/978-3-540-76725-1_25
5. Bianconi, F., Fernández, A., González, E., Caride, D., Calviño, A.: Rotation-invariant colour texture classification through multilayer CCR. Pattern Recognition Letters **30**(8), 765–773 (2009). DOI 10.1016/j.patrec.2009.02.006
6. Bianconi, F., Harvey, R., Southam, P., Fernández, A.: Theoretical and experimental comparison of different approaches for color texture classification. Journal of Electronic Imaging **20**(4), 043,006–1–17 (2011). DOI 10.1117/1.3651210
7. Bianconi, F., Fernández, A.: On the occurrence probability of local binary patterns: a theoretical study. Journal of Mathematical Imaging and Vision **40**(3), 259–268 (2011). DOI 10.1007/s10851-011-0261-7
8. Bonn BTF database (2003). Available online at `http://cg.cs.uni-bonn.de/en/projects/btfdbb/download/`
9. Bradley, A., Jackway, P., Lovell, B.: Classification in scale-space: applications to texture analysis. In: Proceedings of the 14th International Conference on Information Processing in Medical Imaging (IMPI'95), pp. 375–376. Ile de Berder, France (1995)
10. Brodatz, P.: Textures: a photographic album for artists and designers. Dover Publications (1966)
11. Caputo, B., Hayman, E., Mallikarjuna, P.: Class-specific material categorisation. In: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05), vol. II, pp. 1597–1604 (2005)

12. Carstensen, J.M.: Cooccurrence feature performance in texture classification. In: Proceedings of the 8th Scandinavian Conference on Image Analysis (SCIA 1993), pp. 831–838. Tromsø, Norway (1993)
13. Chang, C.I., Chen, Y.: Gradient texture unit coding for texture analysis. Optical Engineering **43**(8), 1891–1902 (2004)
14. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2**, 27:1–27:27 (2011). Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
15. Chen, Y., Chang, C.I.: A new application of texture unit coding to mass classification for mammograms. In: Proceedings of IEEE International Conference on Image Processing, 2004 (ICIP'04), vol. 5, pp. 3335–3338 (2004). DOI 10.1109/ICIP.2004.1421828
16. Crosier, M., Griffin, L.D.: Using basic image features for texture classification. International Journal of Computer Vision **88**, 447–460 (2010). DOI 10.1007/s11263-009-0315-0
17. Daugman, J.: Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. Journal of the Optical Society of America A **2**, 1160–1169 (1985)
18. Davies, E.R.: Introduction to texture analysis. In: M. Mirmehdi, X. Xie, J. Suri (eds.) Handbook of texture analysis, pp. 1–31. Imperial College Press (2008)
19. Dell'Acqua, F., Gamba, P.: Discriminating urban environments using multiscale texture and multiple SAR images. International Journal of Remote Sensing **17**(18), 3797–3812 (2006)
20. Feller, W.: An Introduction to Probability Theory and its Applications, vol. I, 3rd edn. John Wiley & Sons, Inc., New York (1968)
21. Fernández, A., Ghita, O., González, E., Bianconi, F., Whelan, P.F.: Evaluation of robustness against rotation of LBP, CCR and ILBP features in granite texture classification. Machine Vision and Applications **22**(6), 913–926 (2011). DOI 10.1007/s00138-010-0253-4
22. Fernández, A., Álvarez, M.X., Bianconi, F.: Image classification with binary gradient contours. Optics and Lasers in Engineering **49**(9-10), 1177–1184 (2011). DOI 10.1016/j.optlaseng.2011.05.003
23. Fröba, B., Ernst, A.: Face detection with the modified census transform. In: Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition (FGR 2004), pp. 91–96 (2004)
24. Fu, X., Wei, W.: Centralized binary patterns embedded with image euclidean distance for facial expression recognition. In: Proceedings of the Fourth International Conference on Natural Computation (ICNC'08), vol. 4, pp. 115–119 (2008)
25. Ghita, O., Ilea, D.E., Fernández, A., Whelan, P.F.: Local binary patterns versus signal processing texture analysis. A study from a performance evaluation perspective. Sensor Review **32**, 149–162 (2012). DOI 10.1108/02602281211209446
26. Gong, P., Marceau, D.J., Howarth, P.J.: A comparison of spatial feature extraction algorithms for land-use classification with SPOT HRV data. Remote Sensing of Environment **40**, 137–151 (1992)
27. Gool, L.v., Dewaele, P., Oosterlinck, A.: Texture analysis anno 1983. Computer Vision, Graphics, and Image Processing **29**(3), 336–357 (1985). DOI 10.1016/0734-189X(85)90130-6
28. Gotlieb, C.C., Kreyszig, H.E.: Texture descriptors based on co-occurrence matrices. Computer Vision, Graphics, and Image Processing **51**(1), 70–86 (1990). DOI 10.1016/S0734-189X(05)80063-5
29. Griffin, L., Lillholm, M.: Hypotheses for image features, icons and textons. International Journal of Computer Vision **70**, 213–230 (2006)
30. Griffin, L.D.: Private communication (2012)
31. Guo, Z., Zhang, L., Zhang, D.: A completed modeling of local binary pattern operator for texture classification. IEEE Transactions on Image Processing **19**(6), 1657–1663 (2010)
32. Guo, Z., Lin, Q., You, J., Zhang, D., Liu, W.: Local directional derivative pattern for rotation invariant texture classification. Neural Computing & Applications pp. 1–12 (2011). DOI 10.1007/s00521-011-0586-6. Published online 19 April 2011
33. Hafiane, A., Seetharaman, G., Zavidovique, B.: Median binary pattern for textures classification. In: Proceedings of the 4th International Conference on Image Analysis and Recognition (ICIAR 2007), *Lecture Notes in Computer Science*, vol. 4633, pp. 387–398. Montreal, Canada (2007)
34. Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics **3**(6), 610–621 (1973)
35. Harwood, D., Ojala, T., Pietikäinen, M., Kelman, S., Davis, L.S.: Texture classification by center-symmetric auto-correlation, using Kullback discrimination of distributions. Technical Report CAR-TR-678, Center for Automation Research, University of Maryland (1993)
36. Hauth, M., Etzmuss, O., Eberhardt, B., Klein, R., Sarlette, R., Sattler, M., Daubert, K., Kautz, J.: Cloth animation and rendering. In: Proceedings of Eurographics 2002 Tutorials. The Eurographics Association (2002)
37. Hayman, E., Caputo, B., Fritz, M., Eklundh, J.O.: On the significance of real-world conditions for material classification. In: Proceedings of the 8th European Conference on Computer Vision (ECCV 2004), *Lecture Notes in Computer Science*, vol. 3024, pp. 253–266. Springer, Prague, Czech Republic (2004)
38. He, D.C., Wang, L.: Texture unit, texture spectrum, and texture analysis. IEEE Transactions on Geoscience and Remote Sensing **28**(4), 509–512 (1990)
39. He, D.C., Wang, L.: Texture features based on texture spectrum. Pattern Recognition **24**(5), 391–399 (1991)
40. He, D.C., Wang, L.: Unsupervised textural classification of images using the texture spectrum. Pattern Recognition **25**(3), 247–255 (1992)
41. He, Y., Sang, N.: Robust illumination invariant texture classification using gradient local binary patterns. In: Proceedings of 2011 International Workshop on Multi-Platform/Multi-Sensor Remote Sensing and Mapping, pp. 1–6. Xiamen, China (2011)
42. Heikkilä, M., Pietikäinen, M., Schmid, C.: Description of interest regions with center-symmetric local binary patterns. In: P.K. Kalra, S. Peleg (eds.) Proceedings of the 5th Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP 2006), *Lecture Notes in Computer Science*, vol. 4338, pp. 58–69. Madurai, India (2006)
43. Heikkilä, J., Ojansivu, V., Rahtu, E.: Improved blur insensitivity for decorrelated local phase quantization. In: Proceedings of the 20th International Conference on Pattern Recognition (ICPR 2010), vol. 5099, pp. 818–821 (2010)

44. HEP (2012). Code, data and results related to this paper. Available online at http://webs.uvigo.es/antfdez/downloads.html/
45. Hepplewhite, L., Stonhamm, T.J.: Texture classification using N-tuple pattern recognition. In: Proceedings of the 13th International Conference on Pattern Recognition (ICPR'96), vol. 4, pp. 159–163 (1996). DOI 10.1109/ICPR.1996.547253
46. Huang, Y., Wang, Y., Tan, T.: Combining statistics of geometrical and correlative features for 3D face recognition. In: Proceedings of the 17th British Machine Vision Conference, pp. 879–888 (2006)
47. Huang, D., Ardabilian, M., Wang, Y., Chen, L.: A novel geometric facial representation based on multi-scale extended local binary patterns. In: Proceedings of the 9th IEEE International Conference on Automatic Face and Gesture Recognition and Workshops (FG 2011), pp. 1–7 (2011)
48. Huang, D., Shan, C., Ardabilian, M., Wang, Y., Chen, L.: Local binary patterns and its application to facial image analysis: A survey. IEEE Transactions on Systems Man and Cybernetics – Part C: Applications and Reviews **41**(6), 765–781 (2011)
49. Jerry Wu database (2003). Available online at http://www.macs.hw.ac.uk/texturelab/resources/databases/jwdb/
50. Jin, H., Liu, Q., Lu, H., Tong, X.: Face detection using improved LBP under bayesian framework. In: Proceedings of the 3rd International Conference on Image and Graphics, pp. 306–309 (2004). DOI 10.1109/ICIG.2004.62
51. Jin, H., Liu, Q., Tang, X., Lu, H.: Learning local descriptors for face detection. In: Proceedings of IEEE International Conference on Multimedia and Expo, pp. 928–931. IEEE Computer Society (2005). DOI 10.1109/ICME.2005.1521576
52. Junding, S., Shisong, Z., Xiaosheng, W.: Image retrieval based on an improved CS-LBP descriptor. In: Proceedings of the 2nd IEEE International Conference on Information Management and Engineering (ICIME), pp. 115–117. Chengdu, China (2010)
53. Jurie, F., Triggs, B.: Creating efficient codebooks for visual recognition. In: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05), vol. 1, pp. 604–610 (2005)
54. Kandaswamy, U., Schuckers, S.A., Adjeroh, D.: Comparison of texture analysis schemes under nonideal conditions. IEEE Transactions on Image Processing **20**(8), 2260–2275 (2011)
55. Kim, K.I., Jung, K., Park, S.H., Kim, H.J.: Support vector machines for texture classification. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(11), 1542–1550 (2002)
56. Kirsanova, E.N., Sadovsky, M.G.: Entropy approach in the analysis of anisotropy of digital images. Open Systems and Information Dynamics **84**, 239–250 (2002)
57. KTH-TIPS and KTH-TIPS2 databases (2004). Available online at <http://www.nada.kth.se/cvap/databases/kth-tips/
58. Kung, S.Y., Taur, J.S.: Decision-based neural networks with signal/image classification applications. IEEE Transactions on Neural Networks **6**(1), 170–181 (1995). DOI 10.1109/72.363439
59. Kunttu, I., Lepistö, L., Rauhamaa, J., Visa, A.: Image retrieval without segmentation. In: Proceedings of the 10th Finnish Artificial Intelligence Conference, pp. 164–169 (2002)
60. Kunttu, I., Lepistö, L., Rauhamaa, J., Visa, A.: Binary co-occurrence matrix in image database indexing. In: Proceedings of the 13th Scandinavian Conference on Image Analysis (SCIA 2003), *Lecture Notes in Computer Science*, vol. 2749, pp. 1090–1097. Springer, Halmstad, Sweden (2003)
61. Kurmyshev, E.V., Cervantes, M.: A quasi-statistical approach to digital binary image representation. Revista Mexicana de Física **42**(1), 104–116 (1996)
62. Kurmyshev, E.V., Guillén-Bonilla, J.T.: Complexity reduced coding of binary pattern units in image classification. Optics and Lasers in Engineering **49**(6), 718–722 (2011)
63. Lategahn, H., Gross, S., Stehle, T., Aach, T.: Texture classification by modeling joint distributions of local patterns with Gaussian mixtures. IEEE Transactions on Image Processing **19**(6), 1548–1557 (2010)
64. Lazebnik, S., Schmid, C., Ponce, J.: A sparse texture representation using local affine regions. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(8), 1265–1278 (2005)
65. LBP-Bibliography. Available online at http://www.ee.oulu.fi/mvg/page/lbp_bibliography/
66. Lee, A.B., Pedersen, K.S., Mumford, D.: The nonlinear statistics of high-contrast patches in natural images. International Journal of Computer Vision **54**(1-3), 83–103 (2003)
67. Li, S., Kwok, J.T., Zhu, H., Wang, Y.: Texture classification using the support vector machines. Pattern Recognition **36**, 2883–2893 (2003)
68. Liao, S., Law, M.W.K., Chung, A.C.S.: Dominant local binary patterns for texture classification. IEEE Transactions on Image Processing **18**(5), 1107–1118 (2009)
69. Lindsey, C.S., Strömberg, M.: Image classification using the frequencies of simple features. Pattern Recognition Letters **21**, 265–268 (2000)
70. Liu, L., Zhao, L., Longa, Y., Kuanga, G., Fieguth, P.: Extended local binary patterns for texture classification. Image and Vision Computing **30**(2), 86–99 (2012). DOI 10.1016/j.imavis.2012.01.001
71. Madrid-Cuevas, F.J., Medina, R., Prieto, M., Fernández, N.L., Carmona, A.: Simplified texture unit: A new descriptor of the local texture in gray-level images. In: F.J.P. López, A.C. Campilho, N.P. de la Blanca, A. Sanfeliu (eds.) Pattern Recognition and Image Analysis, Proceedings of the First Iberian Conference (IbPRIA 2003), *Lecture Notes in Computer Science*, vol. 2652, pp. 470–477. Springer (2003)
72. Mondial Marmi database (2011). Available online at http://dismac.dii.unipg.it/mm/ver_1_1/index.html/
73. Montgomery, D.C., Runger, G.C.: Applied Statistics and Probability for Engineers, second edn. John Wiley & Sons (1999)
74. Nanni, L., Lumini, A., Brahnam, S.: Local binary patterns variants as texture descriptors for medical image analysis. Artificial Intelligence in Medicine **49**(2), 117–125 (2010)
75. Nanni, L., Brahnam, S., Lumini, A.: A local approach based on a Local Binary Patterns variant texture descriptor for classifying pain states. Expert Systems with Applications **37**(12), 7888–7894 (2010)
76. Nanni, L., Brahnam, S., Lumini, A.: Selecting the best performing rotation invariant patterns in local binary/ternary patterns. In: Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV'10), pp. 369–375. Las Vegas, USA (2010)

77. Nanni, L., Brahnam, S., Lumini, A.: Survey on LBP based texture descriptors for image classification. Expert Systems with Applications **39**(3), 3634–3641 (2012)
78. Nixon, M., Aguado, A.: Feature Extraction & Image Processing. Academic Press, Oxford, UK (2008)
79. Ohanian, P.P., Dubes, R.C.: Performance evaluation for four classes of textural features. Pattern Recognition **25**(8), 819–833 (1992)
80. Oja, E., Valkealahti, K.: Co-occurrence map: Quantizing multidimensional texture histograms. Pattern Recognition Letters **17**(7), 723–730 (1996)
81. Ojala, T., Pietikäinen, M., Harwood, D.: Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In: Proceedings of the 12th IAPR International Conference on Pattern Recognition, vol. 1, pp. 582–585 (1994). DOI 10.1109/ICPR.1994.576366
82. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on feature distributions. Pattern Recognition **29**(1), 51–59 (1996)
83. Ojala, T., Pietikäinen, M., Kyllönen, J.: Gray level cooccurrence histograms via learning vector quantization. In: Proceedings of the 11th Scandinavian Conference on Image Analysis (SCIA 1999), pp. 103–108. Kangerlussuaq, Greenland (1999)
84. Ojala, T., Valkealahti, K., Oja, E., Pietikäinen, M.: Texture discrimination with multidimensional distributions of signed gray level differences. Pattern Recognition **34**, 727–739 (2001)
85. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(7), 971–987 (2002)
86. Ojala, T., Pietikäinen, M., Mäenpää, T., Viertola, J., Kyllönen, J., Huovinen, S.: Outex - new framework for empirical evaluation of texture analysis algorithms. In: Proceedings of the 16th International Conference on Pattern Recognition (ICPR'02), vol. 1, pp. 701–706. IEEE Computer Society, Quebec, Canada (2002)
87. Ojansivu, V., Heikkilä, J.: Blur insensitive texture classification using local phase quantization. In: Proceedings of the International Conference on Image and Signal Processing (ICISP 2008), *Lecture Notes in Computer Science*, vol. 5099, pp. 236–243. Springer (2008)
88. OuTeX database (2002). Available online at `http://www.outex.oulu.fi/`
89. Paclík, P., Verzakov, S., Duin, R.P.W.: Improving the maximum-likelihood co-occurrence classifier: A study on classification of inhomogeneous rock images. In: Proceedings of the 14th Scandinavian Conference on Image Analysis (SCIA 2005), *Lecture Notes in Computer Science*, vol. 3540, pp. 998–1008. Springer (2005)
90. Patel, D., Stonham, T.J.: A single layer neural network for texture discrimination. In: IEEE International Symposium on Circuits and Systems, 1991, vol. 5, pp. 2656–2660 (1991). DOI 10.1109/ISCAS.1991.176092
91. Patel, D., Stonham, T.J.: Texture image classification and segmentation using rank-order clustering. In: Proceedings of the 11th International Conference on Pattern Recognition (ICPR'92), vol. 3, pp. 92–95. IEEE Computer Society (1992)
92. Patel, D., Stonham, T.J.: Unsupervised/supervised texture segmentation and its application to real-world data. In: P. Maragos (ed.) Visual Communications and Image Processing '92, *Proceedings of SPIE*, vol. 1818, pp. 1206–1217. SPIE, Boston, MA, USA (1992). DOI 10.1117/12.131392
93. Patel, D., Stonham, T.J.: Segmentation of potash mine images using multi-layer perceptron networks. In: Proceedings of the Second International Conference on Automation, Robotics and Computer Vision (ICARCV'92). Singapore (1992)
94. Petrou, M., García Sevilla, P.: Image Processing. Dealing with Texture. Wiley Interscience (2006)
95. Pietikäinen, M., Ojala, T., Nisula, J., Heikkinen, J.: Experiments with two industrial problems using texture classification based on feature distributions. In: Proceedings of SPIE, vol. 2354, pp. 197–204 (1994)
96. Qian, X., Hua, X.S., Cheng, P., Ke, L.: PLBP: An effective local binary patterns texture descriptor with pyramid representation. Pattern Recognition **44**, 2502–2515 (2011)
97. Rajpoot, K.M., Rajpoot, N.M.: Wavelets and support vector machines for texture classification. In: Proceedings of 8th International Multitopic Conference (INMIC 2004), pp. 328–333 (2004)
98. Sabeenian, R.S., Dinesh, P.M.: Texture image classification using gray level weight matrix (GLWM). In: Proceedings of the Second International Conference on Advances in Power Electronics and Instrumentation Engineering (PEIE2011), *Communications in Computer and Information Sciences*, vol. 148, pp. 263–266. Springer-Verlag, Maharashtra, India (2011)
99. Sánchez-Yáñez, R.E., Kurmyshev, E.V., Cuevas, F.J.: A framework for texture classification using the coordinated clusters representation. Pattern Recognition Letters **24**(1-3), 21–31 (2003). DOI 10.1016/S0167-8655(02)00185-X
100. Sebe, N., Lew, M.: Robust Computer Vision. Theory and Applications. Kluwer Academic Publishers, Dordrecht, The Netherlands (2003)
101. Singh, M., Singh, S.: Spatial texture analysis: a comparative study. In: Proceedings of the 16th International Conference on Pattern Recognition (ICPR'02), vol. 1, pp. 676–679. IEEE Computer Society (2002). DOI 10.1109/ICPR.2002.1044843
102. Smith, G., Burns, I.: Measuring texture classification algorithms. Pattern Recognition Letters **18**(14), 1495–1501 (1997)
103. Sommerville, D.M.Y.: An Introduction to the Geometry of n Dimensions. Methuen & Co., London (1929)
104. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis and Machine Vision, Third Edition. Thomson (2005)
105. Suguna, R., Anandhakumar, P.: Multi-level local binary pattern analysis for texture characterization. International Journal of Computer Science and Network Security **10**(4), 375–386 (2011)
106. Sun, H., Wang, C., Wang, B., El-Sheimy, N.: Pyramid binary pattern features for real-time pedestrian detection from infrared videos. Neurocomputing **74**, 797–804 (2011)
107. Tan, X., Triggs, B.: Enhanced local texture feature sets for face recognition under difficult lighting conditions. In: Analysis and Modelling of Faces and Gestures, *Lecture Notes in Computer Science*, vol. 4778, pp. 168–182. Springer (2007)
108. Tuceryan, M., Jain, A.K.: Texture analysis. In: C.H. Chen, L.F. Pau, P.S.P. Wang (eds.) Handbook of Pattern Recognition and Computer Vision (2nd Edition), pp. 207–248. World Scientific Publishing (1998)

109. UIUCTex texture database (2005). Available online at `http://www-cvr.ai.uiuc.edu/ponce_grp/data/texture_database/`
110. Unser, M.: Sum and difference histograms for texture classification. IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-8**(1), 118–125 (1986)
111. Unser, M.: Local linear transforms for texture measurements. Signal Processing **11**(1), 61–79 (1986)
112. USC-SIPI image database (1977). Available online at `http://sipi.usc.edu/database/`
113. Valkealahti, K., Oja, E.: Reduced multidimensional cooccurrence histograms in texture classification. IEEE Transactions on Pattern Analysis and Machine Intelligence **20**(1), 90–94 (1998)
114. Varma, M., Zisserman, A.: Texture classification: Are filter banks necessary? In: Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03), vol. 2, pp. 691–698 (2003). DOI 10.1109/CVPR.2003.1211534
115. Varma, M., Zisserman, A.: A statistical approach to material classification using image patch exemplars. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(11), 2032–2047 (2009)
116. Vickers, A.L., Modestino, J.W.: A maximum likelihood approach to texture classification. IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-4**(1), 61–68 (1982). DOI 10.1109/TPAMI.1982.4767197
117. VisTex database (2002). Available online at `http://vismod.media.mit.edu/vismod/imagery/VisionTexture/`
118. Wang, L., He, D.C.: A new statistical approach for texture analysis. Photogrammetric Engineering and Remote Sensing **56**(1), 61–66 (1990)
119. Wang, Y., Wei, X., Xiao, S.: LBP texture analysis based on the local adaptive Niblack algorithm. In: Proceedings of the Congress on Image and Signal Processing, 2008 (CISP '08), vol. 2, pp. 777–780 (2008)
120. Weszka, J.S., Dyer, C.R., Rosenfeld, A.: A comparative study of texture measures for terrain classification. IEEE Transactions on Systems, Man and Cybernetics **6**(4), 269–286 (1976)
121. Wiselin Jiji, G., Ganesan, L.: Comparative analysis of colour models for colour textures based on feature extraction. International Journal of Soft Computing **2**, 361–366 (2007)
122. Wiselin Jiji, G., Ganesan, L.: A new approach for unsupervised segmentation. Applied Soft Computing **10**(3), 689–693 (2010)
123. Wiselin Jiji, G.: Colour texture classification for human tissue images. Applied Soft Computing **11**(2), 1623–1630 (2011)
124. Wu, J.: Rotation invariant classification of 3D surface texture using photometric stereo. Ph.D. thesis, Heriot-Watt University (2003)
125. Wu, X., Sun, J.: A brief study on a novel texture spectrum descriptor for material images. Applied Mechanics and Materials **63-64**, 507–510 (2011)
126. Xiaosheng, W., Junding, S.: An effective texture spectrum descriptor. In: Proceedings of the 5th International Conference on Information Assurance and Security,(IAS 2009), vol. 2, pp. 361–364. Xi'an, China (2009)
127. Xie, X., Mirmehdi, M.: A galaxy of texture features. In: M. Mirmehdi, X. Xie, J. Suri (eds.) Handbook of texture analysis, pp. 375–406. Imperial College Press (2008)
128. Xu, B., Gong, P., Seto, E., Spear, R.: Comparison of gray-level reduction and different texture spectrum encoding methods for land-use classification using a panchromatic Ikonos image. Photogrammetric Engineering and Remote Sensing **69**(5), 529–536 (2003)
129. Zabih, R., Woodfill, J.: Non-parametric local transforms for computing visual correspondence. In: Proceedings of the 3rd European Conference on Computer Vision (ECCV 1994), pp. 151–158. Springer-Verlag, Stockholm, Sweden (1994)
130. Zeng, H.: A robust method for local image feature region description. Zidonghua Xuebao/Acta Automatica Sinica **37**(6), 658–664 (2011). In chinese
131. Zhang, J., Marszałek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: A comprehensive study. International Journal of Computer Vision **73**(2), 213–238 (2007)
132. Zhou, H., Wang, R., Wang, C.: A novel extended local-binary-pattern operator for texture analysis. Information Sciences **178**(22), 4314–4325 (2008). DOI 10.1016/j.ins.2008.07.015



Antonio Fernández was born in Vigo, Spain, in 1967. He received the Industrial Engineering degree and the Ph.D. in Industrial Engineering both from the University of Vigo (Spain) in 1993 and 1998, respectively. He held a research fellowship in the Department of Applied Physics, University of Vigo, during the period 1994 through 1998. He was appointed to the Department of Engineering Design, University of Vigo, in 1999, where he is currently full-time Senior Lecturer in Engineering Drawing. He has worked as a visiting researcher at Centre for Research on Optics (Mexico), University of Perugia (Italy), Dublin City University (Ireland) and Computer Vision Centre (Spain). He has published more than 20 scientific articles in peer-reviewed journals on texture analysis and TV holography.

Marcos X. Álvarez received the MEng in Telecommunications Engineering in 2005, the MSc in Environmental Technology in 2009 and the PhD in the same subject in 2011 – all of them from the University of Vigo (Spain). He is currently a post-doctoral fellow in the Department of Natural Resources and Environmental Engineering of the University of Vigo. His research interests cover computer vision, image processing and pattern recognition, with particular focus on texture analysis.

Francesco Bianconi received the MEng in Mechanical Engineering in 1997 from the University of Perugia (Italy) and the PhD in Computer-aided Design in 2001 from a consortium of italian universities. He is currently Assistant Professor of Design Tools & Methods in Industrial Engineering within the Faculty of Engineering of the University of Perugia. He has been visiting researcher at the University of Vigo (Spain) and the University of East Anglia (UK). His research interests include computer vision, image processing and pattern recognition, with specific interest in theory and applications of texture and colour analysis.