Image classification with binary gradient contours

Antonio Fernández[a], Marcos X. Álvarez[a], Francesco Bianconi[b]

[a] *Department of Engineering Design, Universidade de Vigo*
Escola de Enxeñaría Industrial, Campus Universitario, 36310 Vigo, Spain
[b] *Department of Industrial Engineering, Università degli Study di Perugia*
Via G. Duranti 67, 06125 Perugia, Italy

Abstract

In this work we present a new family of computationally simple texture descriptors, referred to as binary gradient contours (BGC). The BGC methodology relies on computing a set of eight binary gradients between pairs of pixels all along a closed path around the central pixel of a 3 × 3 grayscale image patch. We developed three different versions of BGC features, namely single-loop, double-loop and triple-loop. To quantitatively assess the effectiveness of the proposed approach we performed an ensemble of texture classification experiments over ten different datasets. The obtained results make it apparent that the single-loop version is the best performer of the BGC family. Experiments also show that the single-loop BGC texture operator outperforms the well-known LBP. Statistical significance of the achieved accuracy improvement has been demonstrated through the Wilcoxon signed rank test.

Keywords: texture features, BGC, LBP

Email addresses: antfdez@uvigo.es (Antonio Fernández), marcos@uvigo.es (Marcos X. Álvarez), bianco@unipg.it (Francesco Bianconi)

URL: http://webs.uvigo.es/antfdez/ (Antonio Fernández), http://dismac.dii.unipg.it/bianco/ (Francesco Bianconi)

1. Introduction

Texture analysis plays an important role in computer vision and pattern recognition. Tumor detection in medical imaging for computer-aided diagnosis, automated surface inspection for industrial quality control and terrain classification through the analysis of remote sensed imagery are just some of the applications in which textural information can be successfully exploited. Texture analysis techniques have been recently extended to study dynamic events such as recognition of facial expression [1] or monitoring of paint drying process [2]. Texture descriptors are traditionally classified into four categories: statistical, model-based, geometrical and signal processing methods [3, 4]. Among these categories, statistical methods have become very popular, mainly because they provide good accuracy at an affordable computational cost. The rationale behind statistical texture description is that texture can be represented through the joint distribution of pixel intensities in a local neighbourhood. Based on this assumption, a stationary texture image (i.e., an image that contains a single type of texture) could be ideally characterized by means of the probability distribution of the possible grayscale patterns. This probability can be estimated by a histogram that measures the occurrence frequency of the different grayscale patterns throughout the image. To compute such a histogram, the image is scanned by one-pixel steps with a sliding window, and at each window position the bin corresponding to the detected pattern is incremented 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 is overwhelmingly large even for small neighbourhoods. To cope with multidimensional histograms it is useful to partition the feature space into a discrete vocabulary of local features [5]. It has been recently proposed to reduce the dimension of the histogram through unsupervised clustering of grayscale patterns into a dictionary of *textons* [6]. Reported results show that this method achieves high success rates in texture classification experiments. However, clustering has a number of drawbacks: dependency of the texton dictionary upon the texture samples used to train the classifier, influence of parameter tuning on classification accuracy, and large computational overhead (especially when large neighbourhoods are considered). An alternate approach to partition the feature space is through a closed-form mapping [7]. Mapping-based histogram reduction does not have the drawbacks of clustering, since these schemes define a universal vocabulary of textural features, are parameter-free and compute fast. Several mappings have been proposed by diverse research groups [8, 9, 10, 11, 12, 13, 14]. Despite all of these implementations share the same underlying principle, to the best of our knowledge they have not been yet integrated into a general framework. In this paper we present such a unifying framework. Our claim is that these apparently diverging dimensionality reduction schemes can be interpreted as a mapping from the set of grayscale patterns to a set of integer indexes. This mapping induces a partition of the set of grayscale patterns into groups of *equivalent patterns*. Dimensionality reduction is achieved by merging the occurrence frequencies of equivalent patterns into a single histogram bin. We used this mapping-based framework to describe a novel family of texture descriptors, called binary gradient contours (BGC), which consider the binary gradient of the grayscale values along the eight peripheral pixels of a 3×3 window.

In this class of models a texture is described through the occurrence frequency of the resulting binary 8-tuples. The effectiveness of BGC features has been experimentally demonstrated through an ensemble of texture classification experiments. We have found that one out of the three proposed BGC models is more efficient in discriminating texture than the well-known LBP model.

The remaining of the paper is organized as follows. In Section 2 we present a general framework for texture description based on pattern mapping. Section 3 is devoted to describe the novel family of texture descriptors. The purpose of Section 4 is threefold. The first is to compare the proposed features with the closely related local binary pattern (LBP) concept. The second is to introduce some theoretical considerations to justify the efficiency of our approach. The third is to comparatively analyze the characteristics of the texture descriptors considered in this work. Experimental results are shown in Section 5 and Section 6 summarizes the main conclusions that can be drawn from our work.

2. Framework for texture description based on pattern mapping

To describe the proposed framework, we shall begin by defining the notation to be used henceforth. Let $\mathbf{I}$ be a matrix of M rows and N columns representing the raw pixel intensities of an image quantized to G gray-levels, and $I_{m,n} \in \{0, 1, \ldots, G-1\}$ the pixel intensity corresponding to the m-th row and n-th column. We denote by $\mathbf{S}_{m,n}$ a square crop of 3×3 pixels centered at pixel (m, n) of image $\mathbf{I}$:

$$\mathbf{S}_{m,n} = \begin{bmatrix} I_{m-1,n-1} & I_{m-1,n} & I_{m-1,n+1} \\ I_{m,n-1} & I_{m,n} & I_{m,n+1} \\ I_{m+1,n-1} & I_{m+1,n} & I_{m+1,n+1} \end{bmatrix} \tag{1}$$

Without loss of generality we can rename the terms of the equation above in order to remove the dependance on (m, n). Thus, let $\mathbf{S}$ be a matrix representing the pixel intensities of a generic square neighbourhood with support 3×3. Let I_c be the gray-level of the central pixel and I_j the gray-levels of the peripheral pixels ($j \in \{0, 1, \ldots, 7\}$), which are arranged as follows (see Fig. 1(a)):

$$\mathbf{S} = \begin{bmatrix} I_7 & I_6 & I_5 \\ I_0 & I_c & I_4 \\ I_1 & I_2 & I_3 \end{bmatrix} \tag{2}$$

Let us denote by $\mathcal{M}_{3\times3,G}$ the set of all the possible instances defined by Eq. 2. A typical value for G is 2^8 (i.e., pixel intensity is quantized in 256 levels) since the depth of digitization of most commercial imaging devices is 8 bits. It readily follows that in this case the number of different 3×3 grayscale patterns is given by:

$$\#\mathcal{M}_{3\times3,256} = 2^{72}, \tag{3}$$

where # stands for "cardinality of". It emerges from Eq. 3 that the texture description through the joint distribution of pixel intensity over a 3×3 neighbourhood involves a huge feature vector of roughly 4.7×10^{21} components. Suppose that one intends to describe a texture image through the occurrence frequency of 3×3 grayscale patterns. Provided that 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. It is well-known that such extremely sparse, ultra high dimensional histograms provide an unreliable estimation of the underlying distribution and have negligible discriminant power in image description [15]. Moreover, the 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 by decreasing G. However, as the neighbourhood size increases, the number of bins grows exponentially and soon far outweighs the number of datapoints available in a single image with which to populate the histogram. To tackle such ultra high dimensional feature space we propose to partition $\mathcal{M}_{3\times 3,G}$ into groups of patterns. Dimensionality reduction is straightforwardly attained by merging the histogram bins corresponding to patterns belonging to the same group into a single bin. The partition can be adequately formalized through a mapping that assigns each pattern a non-negative integer index that uniquely identifies the group the pattern belongs to:

$$\begin{array}{rrcl} f: & \mathcal{M}_{3\times 3,G} & \longrightarrow & \mathbb{N} \\ & \mathbf{S} & \longmapsto & k = f(\mathbf{S}) \end{array} \tag{4}$$

The function above establishes an equivalence relation in $\mathcal{M}_{3\times 3,G}$, denoted by $\sim$:

$$\mathbf{S}_1 \sim \mathbf{S}_2 \Leftrightarrow f(\mathbf{S}_1) = f(\mathbf{S}_2) \quad \forall \mathbf{S}_1, \mathbf{S}_2 \in \mathcal{M}_{3\times 3,G}. \tag{5}$$

Let $\mathcal{Q}$ be the range of f [16]:

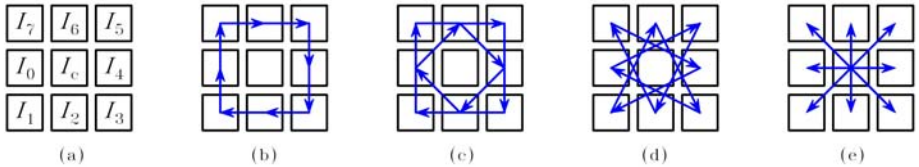


Figure 1: (a) Spatial arrangement of a 3×3 grayscale pattern and schematic representation of the texture models considered in this paper: (b)single-loop, (c) double-loop, and (d) triple-loop versions of the binary gradient contour concept and (e) layout of the well-known local binary pattern.

$$\mathcal{Q} = f(\mathcal{M}_{3\times3,G}), \tag{6}$$

and q the number of different groups of patterns, i.e., the number of equivalence classes:

$$q = \#\mathcal{Q}. \tag{7}$$

The partition can be therefore expressed as:

$$\mathcal{M}_{3\times3,G} = \bigcup_{k\in\mathcal{Q}} \mathcal{M}_{f,k}, \tag{8}$$

where the family of subsets $\{\mathcal{M}_{f,k} \mid k \in \mathcal{Q}\}$ is pairwise disjoint, and each subset is defined by:

$$\mathcal{M}_{f,k} = \{\mathbf{S} \in \mathcal{M}_{3\times3,G} \mid f(\mathbf{S}) = k\}. \tag{9}$$

In the proposed framework, the mapping f makes it possible to represent a texture image $\mathbf{I}$ by a q-dimensional vector $\mathbf{h}_f(\mathbf{I})$ in which the k-th component is given by:

$$h_{f,k}(\mathbf{I}) = \frac{\#\{(m,n) \mid f(\mathbf{S}_{m,n}) = k\}}{(M-2)\times(N-2)}. \tag{10}$$

It is useful to note that in order for sub-image $\mathbf{S}_{m,n}$ to be fully enclosed into $\mathbf{I}$, the crop center cannot be located at the one pixel width periphery of the image, and therefore Eq. 10 must satisfy that $2 \leq m \leq M-1$ and $2 \leq n \leq N-1$.

3. Binary Gradient Contours

We define the binary gradient contour of a 3×3 grayscale image patch as the binary 8-tuple that results of a two-step procedure: 1) the gradient between pairs of pixels is computed along a closed path around the central pixel of $\mathbf{S}$, and 2) the gradients are binarized using the value 0 as a threshold. We propose to define the closed path in three different ways, namely: single-loop, double-loop and triple-loop, as shown in Figs. 1(b)-(d), and the corresponding versions of the binary gradient contours can accordingly be expressed as:

$$\mathbf{g}_1 = \begin{bmatrix} \xi(I_7 - I_0) \\ \xi(I_6 - I_7) \\ \xi(I_5 - I_6) \\ \xi(I_4 - I_5) \\ \xi(I_3 - I_4) \\ \xi(I_2 - I_3) \\ \xi(I_1 - I_2) \\ \xi(I_0 - I_1) \end{bmatrix} \tag{11}$$

$$\mathbf{g}_{2\diamond} = \begin{bmatrix} \xi(I_6 - I_0) \\ \xi(I_4 - I_6) \\ \xi(I_2 - I_4) \\ \xi(I_0 - I_2) \end{bmatrix} \tag{12a}$$

$$\mathbf{g}_{2\square} = \begin{bmatrix} \xi(I_7 - I_1) \\ \xi(I_5 - I_7) \\ \xi(I_3 - I_5) \\ \xi(I_1 - I_3) \end{bmatrix} \tag{12b}$$

$$\mathbf{g}_{2} = \begin{bmatrix} \mathbf{g}_{2\diamond} \\ \mathbf{g}_{2\square} \end{bmatrix} \tag{12c}$$

and

$$\mathbf{g}_{3} = \begin{bmatrix} \xi(I_5 - I_0) \\ \xi(I_2 - I_5) \\ \xi(I_7 - I_2) \\ \xi(I_4 - I_7) \\ \xi(I_1 - I_4) \\ \xi(I_6 - I_1) \\ \xi(I_3 - I_6) \\ \xi(I_0 - I_3) \end{bmatrix} \tag{13}$$

where

$$\xi(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \tag{14}$$

It is useful to notice that the 8-tuples and 4-tuples above are actually functions of $\mathbf{S}$, but the explicit dependency on $\mathbf{S}$ has been dropped from Eqs. 11-13 to alleviate notation.

It readily follows that the number of possible binary 8-tuples is $2^8 = 256$. However, the number of different instances of Eqs. 11, 12c and 13 is below this upper limit since the components of these 8-tuples are not completely independent. Indeed, there are intrinsic constraints for the components of the gradient vectors caused by the fact that these gradients are computed all along a closed path. Taking into account the inequality in the definition of $\xi(x)$ it is evident that the eight components of $\mathbf{g}_1$ cannot be 0's simultaneously. Hence, the number of different single-loop binary gradient contours reduces to 255. The same holds for $\mathbf{g}_3$ and therefore there are 255 different triple-loop binary gradient contours. Similar constraints apply to the two separate closed paths considered in the double-loop binary gradient contour model (see Fig. 1(c)). For the sake of clarity it is convenient to split $\mathbf{g}_2$ into two halves: the first 4-tuple is formed by the four most significant bits of $\mathbf{g}_2$ (Eq. 12a) and the second 4-tuple is formed by the four least significant bits (Eq. 12b), which are referred to as $\mathbf{g}_{2\diamond}$ and $\mathbf{g}_{2\square}$, respectively. It follows that the four components of these 4-tuples cannot be 0's simultaneously. Hence the resulting number of different double-loop binary gradient contours is $(2^4 - 1)^2 = 225$.

In the proposed approach all those grayscale patterns which are mapped to the same binary gradient contour are considered to be "equivalent". We hypothesize that texture can be conveniently described by the histogram that quantifies the occurrence of equivalent 3×3 patterns. One can easily define the mappings corresponding to these equivalence relationships by properly particularizing Eq. 4:

$$\mathrm{BGC1}_{3\times 3}(\mathbf{S}) = \mathbf{w}_8^T \mathbf{g}_1 - 1, \tag{15}$$

$$\mathrm{BGC2}_{3\times 3}(\mathbf{S}) = 15\mathbf{w}_4^T \mathbf{g}_{2\diamond} + \mathbf{w}_4^T \mathbf{g}_{2\square} - 16 \tag{16}$$

and

$$\mathrm{BGC3}_{3\times 3}(\mathbf{S}) = \mathbf{w}_8^T \mathbf{g}_3 - 1, \tag{17}$$

where the superscript ^{T} stands for "the transpose of", and $\mathbf{w}_j$ denotes a vector of weighting factors defined by:

$$\mathbf{w}_j^T = \left[\begin{array}{cccccc} 2^{j-1} & 2^{j-2} & \cdots & 2^1 & 2^0 \end{array}\right] \tag{18}$$

Notice that the weighting vector is chosen arbitrarily. A different sorting of its components would simply result in a different coding of the binary gradient contours. One can observe from Eqs. 15-17 that the different BGC mappings return an integer index between 0 and $q-1$. These indexes are used to label the histogram bins corresponding to each subset of equivalent patterns.

We used matricial notation to define the mappings in a compact form, but Eqs. 15-17 can be rewritten in an extended, algorithmic fashion, as shown in Table 1.

Table 1: Particularization of Eq. 4 for the considered texture models.

$\mathrm{BGC1}_{3\times 3}(\mathbf{S}) = \sum_{n=0}^{7} \xi(I_n - I_{(n+1) \bmod 8}) \times 2^n - 1$
$\mathrm{BGC2}_{3\times 3}(\mathbf{S}) = 15 \times \sum_{n=0}^{3} \xi(I_{2n} - I_{2(n+1) \bmod 8}) \times 2^n + \sum_{n=0}^{3} \xi(I_{2n+1} - I_{(2n+3) \bmod 8}) \times 2^n - 16$
$\mathrm{BGC3}_{3\times 3}(\mathbf{S}) = \sum_{n=0}^{7} \xi(I_{3n \bmod 8} - I_{3(n+1) \bmod 8}) \times 2^n - 1$
$\mathrm{LBP}_{3\times 3}(\mathbf{S}) = \sum_{n=0}^{7} \xi(I_n - I_c) \times 2^n$

4. Discussion

This section is devoted to discuss some important points of the proposed family of texture descriptors. We shall perform a comparative analysis of the three different versions of BGC features proposed in this paper. In addition we shall also study the differences and similarities between LBP and BGC models.

4.1. Relationship with LBP

The LBP model is a well-known, widely used approach to texture description. As a consequence of the large attention received by the LBP methodology from the texture research community, a vast body of literature reporting on different applications of LBP features is currently available [17]. Such a success relies on three major reasons: 1) LBP model is conceptually simple, 2) LBP method is not too demanding from a computational point of view, being well suited for real-time processing, and 3) LBP features achieve high texture discrimination accuracy. Detailed descriptions of the LBP model can be found elsewhere, for instance in [18]. The $\mathrm{LBP}_{3\times 3}$ texture operator is usually defined through the concept of local thresholding: the grayscale values of the periphery of a 3×3 window are converted into a set of binary values using the grayscale value of the central pixel as a threshold. The LBP assigns each image pixel a code which is commonly defined by the formula shown in Table 1.

Herein we propose to regard the LBP model from a different perspective. Indeed the LBP can be interpreted as a mapping from the grayscale pattern space to the 8-tuple binary space, and therefore this texture model perfectly fits into the general framework described in Sec. 2. In order to better highlight the adequacy of the mapping-based framework to describe the LBP model, it is convenient to reformulate the LBP using the notation introduced in Sec. 2. To this end we define the following 8-tuple:

$$\mathbf{g}_{\text{✺}} = \begin{bmatrix} \xi(I_7 - I_c) \\ \xi(I_6 - I_c) \\ \xi(I_5 - I_c) \\ \xi(I_4 - I_c) \\ \xi(I_3 - I_c) \\ \xi(I_2 - I_c) \\ \xi(I_1 - I_c) \\ \xi(I_0 - I_c) \end{bmatrix} \tag{19}$$

It should be noticed that the dependency on $\mathbf{S}$ has been removed for the sake of simplicity, following an analogous approach to that of Eqs. 11-13. The equation above represents the eight binary gradients at the central pixel, calculated along the orientations determined by the eight neighbouring pixels and binarized using the value 0 as a threshold. There are $2^8 = 256$ different instances of Eq. 19, since in this case intensity gradients are not computed along a closed path and therefore any component of the 8-tuple can take the values 0 and 1, irrespective of the values of the other components.

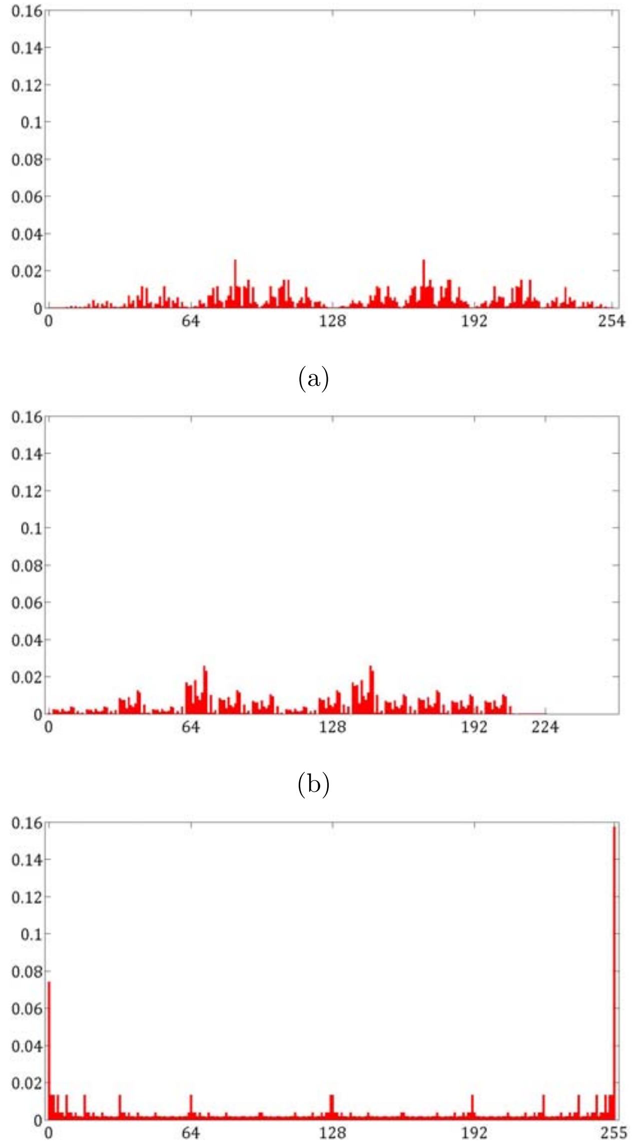
In the LBP texture model the grayscale patterns that are assigned the same 8-tuple defined in Eq. 19 belong to the same group of equivalent patterns. Texture is then represented through a 256 bin histogram which quantifies the occurrence frequency of the different groups of equivalent patterns. Each group of patterns can be codified for labeling purposes using the previously defined matricial notation:

$$\mathrm{LBP}_{3\times3}(\mathbf{S}) = \mathbf{w}_8^T \mathbf{g}_{\text{❊}} \tag{20}$$

Notice that in contrast to Eqs. 15-17, in this case it is not necessary to include a subtractive term into the formula since the escalar product of $\mathbf{w}_8^T$ and $\mathbf{g}_{\text{❊}}$ can take the value 0.

4.2. Theoretical efficiency

The question at this point is: provided that LBP features are very effective in discriminating textures, why should we use the closely related BGC features? We argue that the a priori advantage of using BGC rather than LBP features can be theoretically justified through considerations on information efficiency. The sets of equivalence classes used to describe texture can be regarded to as alphabets of symbols, in analogy with a discrete noiseless channel [19]. We observe that the dimension of the feature space represents a theoretical limit to the amount of information that can be conveyed through a texture model. It is well-known that the highest efficiency attainable by an alphabet occurs when its symbols are equally likely, that is to say when the entropy of the histogram representing the probability distribution of each



(a)

(b)

(c)

Figure 2: A priory probabilities $P_f(k)$ of the equivalence classes corresponding to the following mappings: (a) $\mathrm{BGC1}_{3\times3}$ and $\mathrm{BGC3}_{3\times3}$ (b) $\mathrm{BGC2}_{3\times3}$, and (c) $\mathrm{LBP}_{3\times3}$. The histograms were computed for $G = 12$.

symbol is maximum [19]. Therefore we define the efficiency e_f of a texture model in the following way:

$$e_f = \frac{\sum\limits_{k=0}^{q-1} P_f(k) \log_2 P_f(k)}{\log_2 q}, \tag{21}$$

where $P_f(k)$ is the a priori probability of the k-th equivalence class of the texture model defined by mapping f.

The numerator of Eq. 21 represents the actual entropy (i.e. the amount of information) of the alphabet, and the denominator the maximum attainable entropy (under the assumption of equiprobable symbols). We conjecture that the effectiveness of a texture description method is related to its efficiency. The dimensionality of the considered texture models are listed in Table 2 as well as the maximum attainable entropy, which would correspond to a flat histogram.

In order to estimate the actual entropy of the considered models, we have empirically determined $P_f(k)$ by calculating the proportion of 3×3 grayscale patterns that belong to each equivalence class, assuming that all the possible 3×3 grayscale patterns are equally likely. To this end, we have computer generated all the possible 3×3 grayscale patterns, then we have mapped each pattern to its corresponding code through Eqs. 15-17 and 20, and finally we have divided the number of patterns belonging to each equivalence class by the total number of patterns (G^9):

$$P_f(k) = \frac{\#\mathcal{M}_{f,k}}{G^9} \tag{22}$$

The a priori probabilities of the equivalence classes defined through BGC

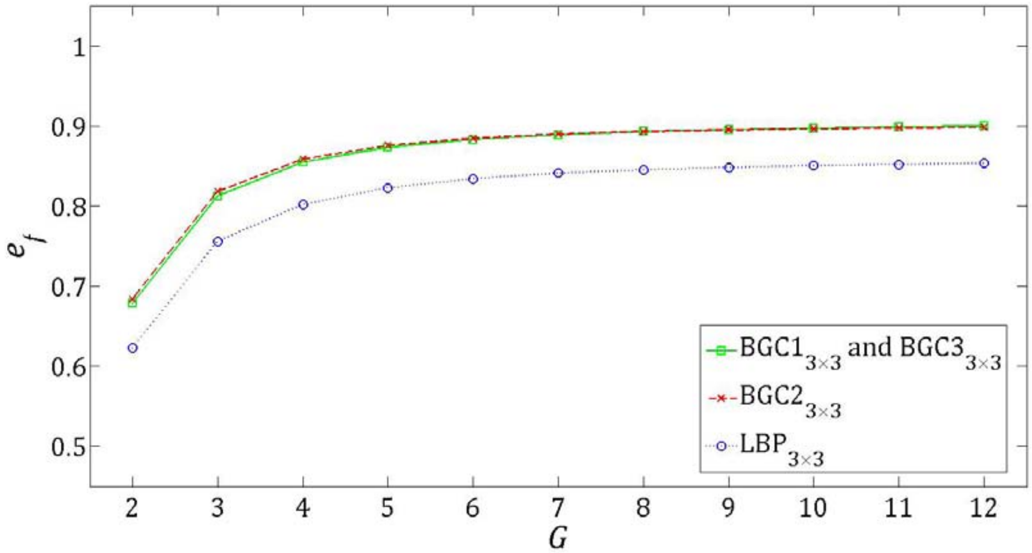


Figure 3: Dependency of the theoretical efficiency of the considered texture models with the number of gray-levels.

and LBP mappings are shown in Fig. 2. Notice that the plots corresponding to BGC1$_{3\times3}$ and BGC3$_{3\times3}$ models are identical. This coincidence will be explained in Sect. 4.3. Efficiency was computed by applying Eq. 21 to these a priori probabilities. We estimated $P_f(k)$ only for moderate values of G (number of quantization levels), since as G increases, the number of possible 3×3 grayscale patterns grows exponentially and computing becomes time consuming. Anyway, the considered range of G seems sufficient, since efficiency converges for $G > 7$, as shown in Fig. 3.

4.3. Comparative analysis

Although conceptually identical, the three members of the proposed BGC family of texture features exhibit some differences which are worthy of comment. Besides, it is fruitful to analyze the likenesses and divergences between

Table 2: Summary of characteristics of the considered texture models: dimensionality of the feature space (q), maximum attainable entropy expressed in bits ($\log_2 q$), theoretical efficiency (e_f) computed for $G = 12$ quantization levels, separation between pixels (Δ), and gradient orientations (θ).

Model	q	$\log_2 q$	e_f	Δ	θ
$\mathrm{BGC1}_{3\times 3}$	255	7.9944	0.8996	1	$0, \pm\frac{\pi}{2}, \pi$
$\mathrm{BGC2}_{3\times 3}$	225	7.8138	0.8983	$\sqrt{2}, 2$	$0, \pm\frac{\pi}{4}, \pm\frac{\pi}{2}, \pm\frac{3\pi}{4}, \pi$
$\mathrm{BGC3}_{3\times 3}$	255	7.9944	0.8996	$\sqrt{5}$	$\pm\arctan\frac{1}{2}, \pi \pm \arctan\frac{1}{2}, \pm\arctan 2, \pi \pm \arctan 2$
$\mathrm{LBP}_{3\times 3}$	256	8	0.8533	$1, \sqrt{2}$	$0, \pm\frac{\pi}{4}, \pm\frac{\pi}{2}, \pm\frac{3\pi}{4}, \pi$

BGC and LBP texture descriptors.

An important property of both BGC and LBP models is intensity invariance. Indeed, these descriptors are insensitive with respect to monotonic transforms of the grayscale values of image pixels, as one can readily ascertain from definitions.

All the considered models are based on binary 8-tuples. However, the corresponding feature spaces have different dimension. The $\mathrm{LBP}_{3\times 3}$ histogram has $2^8 = 256$ bins, but the BGC histograms do not reach this upper limit since the components of the 8-tuples are subject to intrinsic constraints discussed in Sec. 3. The dimensions of these feature spaces are summarized in Table 2. The fact that the models have different dimensions determines that the maximum attainable entropies are also different.

Some interesting remarks can be made based on the analysis of theoretical efficiency performed in Sec. 4.2. First, it clearly emerges from Fig. 2 that the a priori probability of the equivalence classes defined through the corresponding mappings are rather different. The $\mathrm{LBP}_{3\times 3}$ model exhibits two conspicuous peaks located at both extremes of the histogram, while in the BGC histograms the occurrence frequencies are more evenly distributed. Concretely, 23.14% of the possible grayscale 3×3 patterns concentrate on just two of the subsets in which the set $\mathcal{M}_{3\times3,12}$ is partitioned through the $\mathrm{LBP}_{3\times3}$ mapping. The remaining 76.86% patterns distribute in turn throughout the other 254 subsets of patterns. Conversely, the BGC models produce a more balanced partition of $\mathcal{M}_{3\times3,12}$ since in these cases the largest subset contains only 2.57% of the possible grayscale patterns. Second, it is important to realize that the a priori probabilities of the equivalence classes induced by $\mathrm{BGC1}_{3\times3}$ and $\mathrm{BGC3}_{3\times3}$ models are exactly the same. This is motivated, on the one hand, by the fact that both binary gradients are computed all along a unique closed path instead of two separate closed paths as in $\mathrm{BGC2}_{3\times3}$ model (see Fig. 1), and on the other hand, by the hypothesis that pixel intensities are statistically independent. And third, one can conclude from Fig. 3 that BGC mappings convey a higher amount of textural information than $\mathrm{LBP}_{3\times3}$.

The schematic representation depicted in Fig. 1 makes it apparent that gradient scale and orientation vary from one model to another. Thus, the sampling interval Δ is unique in $\mathrm{BGC1}_{3\times3}$ and $\mathrm{BGC3}_{3\times3}$ operators (see Table 2) while grayscale pixel values are pairwise compared at two different scales in $\mathrm{BGC2}_{3\times3}$ and $\mathrm{LBP}_{3\times3}$ operators. With regard to orientation, gradients are computed at four different angles in the $\mathrm{BGC1}_{3\times3}$ model while eight different angles are considered in the other models. As a consequence, the $\mathrm{BGC1}_{3\times3}$ model encompasses only four types of pairwise pixel interactions, namely the one-pixel step gradient at four different orientation, whereas the rest of the models incorporate a wider variety of pairwise pixel interactions.

As a final remark it should be pointed out that the $\text{LBP}_{3\times3}$ operator takes into account all the pixels of the 3×3 neighbourhood, while the BGC models only take into consideration the eight peripheral pixels. Although it could seem somewhat abnormal to discard the central pixel, similar approaches to texture description have been previously reported. The motivation to leave the central pixel out relies on the fact that textures can often be considered realizations of a Markov random field, and in this framework, the probability of the central pixel depends only on its neighbourhood [20]. In the "neighbourhood classifier" [6] texture is modeled through the occurrence frequency of the textons of a dictionary. Textons are the resulting centroids of aggregating and clustering source image patches. These image patches are formed by the raw pixel intensities of a square neighbourhood in which the central pixel is left out, analogously to the BGC model. Another texture model that excludes the central pixel is the "texture co-occurrence spectrum" [21], in which four oriented masks at 0, 45, 90 and 135 degrees around each pixel are used to extract textural information in terms of the occurrence of 4-conjoint pixel values for a given direction. Rank coding was used to reduce the dimensionality of the features.

5. Experimental results

In order to assess the validity of the proposed approach we performed a set of texture classification experiments using ten different datasets. Nine of them are routinely used by the texture classification research community, in part because these datasets are available in the Internet. The benchmark is composed of the following datasets: four test suites from the OuTeX data-

Table 3: Properties of the datasets used in the experiments.

Database	Classes	Samples per class	Sample resolution (pixels)	Format	Predefined train/test sets?
OuTeX TC_00000	24	20	128 × 128	grayscale	Yes
OuTeX TC_00001	24	88	64 × 64	grayscale	Yes
OuTeX TC_00002	24	368	32 × 32	grayscale	Yes
OuTeX TC_00013	68	20	128 × 128	Colour	No
Brodatz	13	16	256 × 256	Colour	No
KTH-TIPS	10	4	100 × 100	grayscale	No
KTH-TIPS2	44	4	100 × 100	grayscale	No
VisTex	167	4	256 × 256	Colour	No
Jerry Wu	39	4	256 × 256	grayscale	No
Mondial Marmi	12	4	544 × 544	Colour	No

base, namely TC_00000, TC_00001, TC_00002 and TC_00013 [22]; a set of images generated from the popular Brodatz album, which can be downloaded from the volume "Textures" of the USC-SIPI image database [23]; the set of images from the KTH-TIPS and KTH-TIPS2 databases [24] with the following settings: scale number = 5, object pose = frontal and illumination direction = frontal; the "Reference Textures" of the VisTex database [25]; and a set of images from the Jerry Wu database [26] with the following settings: rotation = 0, slant = 45 and tilt = 0. The tenth dataset, referred to as Mondial Marmi, is composed of images of polished granite surfaces. As one can readily ascertain from Fig. 4, some of the granite classes have very similar visual properties. This dataset has been created by the authors and is thoroughly documented in a previous work [27]. Table 3 summarizes the format and structure details of these ten datasets. In all cases the texture samples have been generated by partitioning the original images into non-overlapping subimages. Colour images have been converted to grayscale

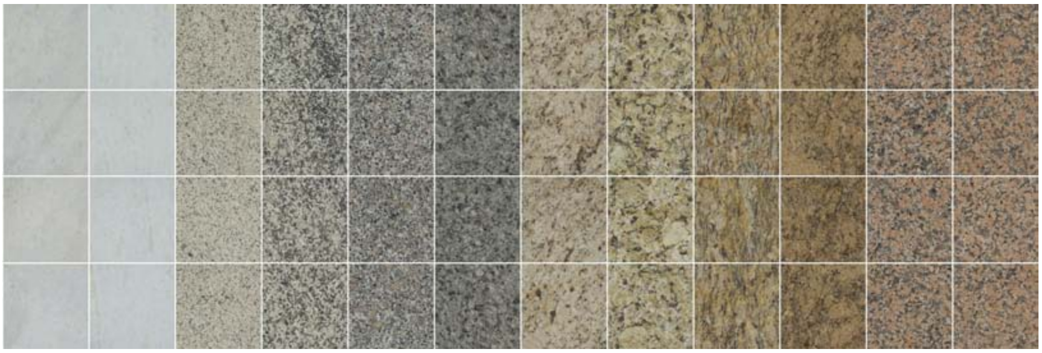


Figure 4: Granite images of the Mondial Marmi dataset. Each column corresponds to a different texture class.

format by means of Matlab's `rgb2gray` function.

The classification experiments were based on the nearest neighbour rule [28] with the L_1 norm, also called Manhattan distance [29]. It is reasonable to expect that efficient classification algorithms, such as AdaBoost, self organizing maps, support vector machines, etc., may yield increased accuracy. The choice of the simple 1-NN classifier is intended to highlight the discriminant power of the texture descriptors rather than to maximize the success rate. The generalization error was estimated through split-half validation [30], i.e., each dataset was randomly divided into two disjoint subsets of the same cardinality, one used for training and the other for testing. We adopted a stratified sampling scheme [31], so that the proportion of samples of each class in the training set is maintained the same as in the overall dataset to avoid class biasing in the classification process. The success rate, i.e., the percentage of correctly classified patterns of the test set, is calculated for 100 different random partitions into training and validation set in order to have a stable estimation of the generalization error.

Table 4: Experimental success rate (mean ± standard deviation): percentage of samples correctly classified averaged over 100 trials.

Database	BGC1$_{3\times 3}$	BGC2$_{3\times 3}$	BGC3$_{3\times 3}$	LBP$_{3\times 3}$
OuTeX TC_00000	99.80 ± 0.32	98.52 ± 0.67	98.83 ± 0.64	99.68 ± 0.40
OuTeX TC_00001	98.12 ± 0.37	95.59 ± 0.43	96.33 ± 0.47	98.44 ± 0.28
OuTeX TC_00002	88.52 ± 0.34	81.04 ± 0.46	80.63 ± 0.41	86.24 ± 0.42
OuTeX TC_00013	79.75 ± 0.97	76.34 ± 1.17	75.94 ± 1.00	78.32 ± 0.88
Brodatz	100.00 ± 0.00	98.00 ± 1.31	99.85 ± 0.35	100.00 ± 0.00
KTH-TIPS	100.00 ± 0.00	99.65 ± 1.28	99.45 ± 1.57	100.00 ± 0.00
KTH-TIPS2	83.48 ± 2.38	74.48 ± 2.92	78.11 ± 2.85	82.26 ± 2.73
VisTex	79.92 ± 2.12	73.86 ± 2.39	74.33 ± 2.33	77.99 ± 2.33
Jerry Wu	97.44 ± 1.34	95.35 ± 2.23	97.97 ± 1.26	97.41 ± 1.42
Mondial Marmi	90.50 ± 4.89	87.42 ± 6.04	89.38 ± 5.11	87.50 ± 5.95

For comparison purposes we performed the classification experiments using BGC1$_{3\times 3}$, BGC2$_{3\times 3}$, BGC3$_{3\times 3}$ and LBP$_{3\times 3}$ texture features. The obtained classification accuracies are gathered in Table 4. One can readily observe that there is a considerable spread in models performance. In order to determine whether or not the differences on accuracy are significant it is usual to perform statistical hypothesis testing [32]. We performed the Wilcoxon signed rank test [33] on all the pairwise combinations of the four texture models considered in this work. The same set of six tests was repeated 10 times, one for each dataset. The results on statistical significance have been collected in a 4×4 matrix (see Table 5), where the value in the i-th row and j-th column represents the number of datasets for which the mean accuracy achieved by the i-th model is significantly higher than the mean accuracy achieved by the j-th model. The sum of a non-diagonal element and its symmetric with respect to the main diagonal is less than or equal to the number of datasets employed (10 in our experiments). Strict inequality

Table 5: Results of the Wilcoxon signed rank test ($\alpha = 5\%$). Matrix elements denote the number of datasets for which the model placed in the corresponding row achieves a significantly higher accuracy than the model placed in the corresponding column.

	$\text{BGC1}_{3\times3}$	$\text{BGC2}_{3\times3}$	$\text{BGC3}_{3\times3}$	$\text{LBP}_{3\times3}$
$\text{BGC1}_{3\times3}$	-	10	9	6
$\text{BGC2}_{3\times3}$	0	-	2	0
$\text{BGC3}_{3\times3}$	1	7	-	2
$\text{LBP}_{3\times3}$	1	9	8	-

holds whenever the differences between a pair of models are not significant for at least one dataset.

With regard to the BGC features, it emerges from Tables 4 and 5 that the $\text{BGC1}_{3\times3}$ model is clearly the best performer of the novel family of texture descriptors. The second place of the BGC ranking pertains to the $\text{BGC3}_{3\times3}$ model. It is worth mentioning that $\text{BGC3}_{3\times3}$ model is markedly less accurate than $\text{BGC1}_{3\times3}$ model although both models have the same theoretical efficiency (see Sec. 4.2 and Fig. 3). The most likely explanation for this contradiction between experimental results and theoretical considerations is that in real textures pixel intensities are correlated to some extent. Hence the basic hypothesis on which the theoretical efficiency analysis is based, namely that pixel intensities are statistically independent, is not completely valid. The double-loop version is the worst performer. To explain its lower discriminative power it is important to note that in the $\text{BGC2}_{3\times3}$ model the binary 8-tuple is formed by concatenating two independent sets of four binary gradients (see Eq. 12c and Fig. 1(c)). As a consequence the $\text{BGC2}_{3\times3}$ model can only encode a four-dimensional joint distribution of pixel intensity, while the BGC1$_{3\times3}$ and BGC3$_{3\times3}$ models encode an eight-dimensional joint distribution.

With regard to the comparison between BGC1$_{3\times3}$ and LBP$_{3\times3}$ features, in seven out of the 10 datasets we can confidently ($\alpha = 5\%$) reject the null hypothesis that the means of the two success rate distributions are identical, whereas in the other three datasets no significant differences arise between both texture models. It also emerges that in six out of the seven datasets in which a significant difference is observed, the BGC1$_{3\times3}$ outperforms the LBP$_{3\times3}$. The contrary occurs with one dataset. In short, the score achieved by the BGC1$_{3\times3}$ texture descriptor in this comparative assessment against the LBP$_{3\times3}$ model is: six wins, three ties and one loss. It should be noted that, on the one hand, in two out of the three ties (Brodatz and KTH-TIPS datasets) perfect classification was achieved by both models, and on the other hand, the accuracy improvement provided by the LBP$_{3\times3}$ in the only case when the BGC$_{3\times3}$ model was defeated (dataset OuTeX TC_00001) is fairly slim.

6. Conclusions

In this work we presented a conceptually simple and computationally efficient family of new texture descriptors. Three different methods have been proposed, namely single-loop, double-loop and triple-loop binary gradient contours, which are based on pairwise comparisons of pixel intensities all along the periphery of a 3×3 window. These models have been comparatively analyzed from a theoretical standpoint. A texture classification experiment conducted over 10 different datasets made it apparent that all the BGC methods have a high texture discrimination capability. We found that the single-loop binary gradient contour model, referred to as BGC$1_{3\times3}$, is the best performer of the BGC family. Comparison with the well-known LBP$_{3\times3}$ texture model showed that BGC$1_{3\times3}$ works better in most cases.

[1] A. Hadid, G. Zhao, T. Ahonen, M. Pietikäinen, Face analysis using local binary patterns, in: M. Mirmehdi, X. Xie, J. Suri (Eds.), Handbook of texture analysis, Imperial College Press, 2008, pp. 347–373.

[2] R. Alves Braga Junior, B. Oliveira Silva, G. Rabelo, R. Marques Costa, A. Machado Enesa, N. Cap, H. Rabal, R. Arizaga, M. Trivi, G. Horgan, Reliability of biospeckle image analysis, Optics and Lasers in Engineering 45 (2007) 390–395.

[3] M. Tuceryan, A. K. Jain, Texture analysis, in: C. H. Chen, L. F. Pau, P. S. P. Wang (Eds.), Handbook of Pattern Recognition and Computer Vision (2nd Edition), World Scientific Publishing, 1998, pp. 207–248.

[4] X. Xie, M. Mirmehdi, A galaxy of texture features, in: M. Mirmehdi, X. Xie, J. Suri (Eds.), Handbook of texture analysis, Imperial College Press, 2008, pp. 375–406.

[5] M. Crosier, L. D. Griffin, Using basic image features for texture classification, International Journal of Computer Vision 88 (2010) 447–460. doi:10.1007/s11263-009-0315-0.

[6] M. Varma, A. Zisserman, A statistical approach to material classification using image patch exemplars, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (11) (2009) 2032–2047.

[7] F. Bianconi, A. Fernández, On the occurrence probability of local binary patterns: a theoretical study, Journal of Mathematical Imaging and Vision 40 (3) (2011) 259–268. doi:10.1007/s10851-011-0261-7.

[8] D.-C. He, L. Wang, Texture unit, texture spectrum, and texture analysis, IEEE Transactions on Geoscience and Remote Sensing 28 (4) (1990) 509–512.

[9] R. Zabih, J. Woodfill, Non-parametric local transforms for computing visual correspondence, in: Proceedings of the 3rd European Conference on Computer Vision (ECCV), Springer-Verlag, Stockholm, Sweden, 1994, pp. 151–158.

[10] S. Y. Kung, J. S. Taur, Decision-based neural networks with signal/image classification applications, IEEE Transactions on Neural Networks 6 (1) (1995) 170–181. doi:10.1109/72.363439.

[11] T. Ojala, M. Pietikäinen, D. Harwood, A comparative study of texture measures with classification based on feature distributions, Pattern Recognition 29 (1) (1996) 51–59.

[12] R. E. Sánchez-Yáñez, E. V. Kurmyshev, F. J. Cuevas, A framework for texture classification using the coordinated clusters representation, Pattern Recognition Letters 24 (1-3) (2003) 21–31. doi:10.1016/S0167-8655(02)00185-X.

[13] F. J. Madrid-Cuevas, R. Medina, M. Prieto, N. L. Fernández, A. Carmona, 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), Vol. 2652 of Lecture Notes in Computer Science, Springer, 2003, pp. 470–477. doi:10.1007/978-3-540-44871-6_55.

[14] H. Jin, Q. Liu, H. Lu, X. Tong, Face detection using improved LBP under bayesian framework, in: Proceedings of the 3rd International Conference on Image and Graphics, 2004, pp. 306–309. doi:10.1109/ICIG.2004.62.

[15] H. Lategahn, S. Gross, T. Stehle, T. Aach, Texture classification by modeling joint distributions of local patterns with Gaussian mixtures, IEEE Transactions on Image Processing 19 (6) (2010) 1548–1557.

[16] E. W. Weisstein, CRC Concise Encyclopedia of Mathematics, 2nd Edition, Chapman & Hall/CRC, 2002.

[17] LBP bibliography. Available online at `<http:// www.ee.oulu.fi/ mvg/ page/ lbp_bibliography/>`.

[18] T. Mäenpää, M. Pietikäinen, Texture analysis with local binary patterns, in: C. H. Chen, P. S. P. Wang (Eds.), Handbook of Pattern Recognition and Computer Vision (3rd Edition), World Scientific Publishing, 2005, pp. 197–216.

[19] R. E. Blahut, Principles and Practice of Information Theory, Addison-Wesley, 1990.

[20] M. Petrou, P. García Sevilla, Image Processing. Dealing with Texture, Wiley Interscience, 2006.

[21] D. Patel, T. J. Stonham, Unsupervised/supervised texture segmentation and its application to real-world data, in: P. Maragos (Ed.), Visual Communications and Image Processing '92, Vol. 1818 of Proceedings of SPIE, SPIE, Boston, MA, USA, 1992, pp. 1206–1217. doi:10.1117/12.131392.

[22] OuTeX database. Available online at <http:// www.outex.oulu.fi/>.

[23] The USC-SIPI image database. Available online at <http:// sipi.usc.edu/ database/>.

[24] KTH-TIPS and KTH-TIPS2 databases. Available online at <http:// www. nada.kth.se/ cvap/ databases/ kth-tips/>.

[25] VisTex database. Available online at <http:// vismod.media.mit.edu/ vismod/ imagery/ VisionTexture/>.

[26] Jerry Wu database. Available online at <http:// www.macs.hw.ac.uk/ texturelab/ resources/ databases/ jwdb/>.

[27] A. Fernández, O. Ghita, E. González, F. Bianconi, P. F. Whelan, Evaluation of robustness against rotation of LBP, CCR and ILBP features in granite texture classification, Machine Vision and Applications (in press, published online 24 February 2010). doi:10.1007/s00138-010-0253-4.

[28] R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification, 2nd Edition, Wiley-Interscience, 2001.

[29] Y. Rubner, J. Puzicha, C. Tomasi, J. M. Buhmann, Empirical evaluation of dissimilarity measures for color and texture, Computer Vision and Image Understanding 84 (2001) 25–43.

[30] E. W. Steyerberg, F. E. Harrell, G. J. J. M. Borsboom, M. J. C. Eijkemans, Y. Vergouwe, J. D. F. Habbema, Internal validation of predictive models: Efficiency of some procedures for logistic regression analysis, Journal of Clinical Epidemiology 54 (8) (2001) 774–781.

[31] U. Braga-Neto, E. Dougherty, Exact performance of error estimators for discrete classifiers, Pattern Recognition 38 (2005) 1799–1814.

[32] P. Southam, R. Harvey, Towards texture classification in real scenes, in: Proceedings of the British Machine Vision Conference, Oxford, UK, 2005, pp. 240–250.

[33] D. C. Montgomery, G. C. Runger, Applied Statistics and Probability for Engineers, 2nd Edition, John Wiley & Sons, 1999.