Pattern Recognition 45 (2012) 4312–4325

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

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

Texton theory revisited: A bag-of-words approach to combine textons

Susana Alvarez [a,*], Maria Vanrell [b,c]

[a] Dept. Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili, Campus Sescelades, Avinguda dels Països Catalans, 26, 43007 Tarragona, Spain
[b] Dept. Ciències de la Computació—Universitat Autònoma de Barcelona, Spain
[c] Computer Vision Center, Edifici O, Campus UAB, 08193 Bellaterra, Barcelona, Spain

ARTICLE INFO

Article history:
Received 24 May 2011
Received in revised form
20 April 2012
Accepted 30 April 2012
Available online 22 May 2012

Keywords:
Colour-texture attributes
Perceptual descriptor
Colour textons

ABSTRACT

The aim of this paper is to revisit an old theory of texture perception and update its computational implementation by extending it to colour. With this in mind we try to capture the optimality of perceptual systems. This is achieved in the proposed approach by sharing well-known early stages of the visual processes and extracting low-dimensional features that perfectly encode adequate properties for a large variety of textures without needing further learning stages.

We propose several descriptors in a bag-of-words framework that are derived from different quantisation models on to the feature spaces. Our perceptual features are directly given by the shape and colour attributes of image blobs, which are the textons. In this way we avoid learning visual words and directly build the vocabularies on these low-dimensional texton spaces. Main differences between proposed descriptors rely on how co-occurrence of blob attributes is represented in the vocabularies. Our approach overcomes current state-of-art in colour texture description which is proved in several experiments on large texture datasets.

1. Introduction

Computational approaches dealing with texture representation in natural images have been one focus of interest since the early days of computer vision research. Different approaches have been proposed from very different points of views [1–3] providing competent results in specific applications or specific datasets. However, there is not a one definite description of texture proved to be consistently applicable to extensive image datasets and sharing at least the early stages of most low-level visual descriptors (such as a multi-scale Laplacian pyramid [4]). Such a description inspired in human perception should be the final aim, since texture description is just pursuing the simulation of a pure human ability.

A second problem arising when dealing with texture description is how to integrate colour in texture representations. It is not yet clear what is the best way to combine these two visual cues [3]. The main reason probably arise from their inherent different spatial nature; while colour is a pixel property, texture is a property of an image region. Usual approaches for feature integration have been mainly twofold. First, colour and texture are processed separately and then combined at the similarity measure level [5–7], this means that for every visual cue a dissimilarity measure is obtained, each one in a different space needing to be scaled in a final similarity measure. In a second approach, colour and texture are jointly processed applying the same texture descriptor over each component of a colour space, and outputs are finally concatenated in order to obtain a feature vector [3,8,9].

In this paper we propose to deal with the two above-mentioned problems by revisiting an old theory of human texture perception and propose this to be extended to colour through the texton concept. We propose a revision of the texton theory [10] by introducing a computational approach that deals with the original definition of texton. Moreover, in the frame of this theory, we propose to integrate colour as one more texton, this is, as one ore attribute of image blobs. In this way we do not propose a completely new descriptor, we propose a new approach to compute image textons following the original definition [10] and adding colour.

The study of how texture perception is dealt by humans [11] has been addressed by finding perceptual representations that correlate with pre-attentive texture segregation [12] or with similarity judgements [13], both given by human observers in psychophysical experiments. After different conjectures, Julesz and Bergen [10] proposed the texton theory, which is summarised in three heuristics. First, *"texture discrimination is a preattentive visual task"*. Second, *"textons are elongated blobs (e.g., rectangles, ellipses, line segments with specific colours, angular orientations, widths and lengths), terminators and crossings"*. Third, *"preattentive vision directs attentive vision to the location where differences in density of textons occur, ignoring positional relationships between*

* Corresponding author. Tel.: +34 977 55 96 93.
E-mail address: susana.alvarez@urv.es (S. Alvarez).

0031-3203/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2012.04.032

textons.'' Some lines below, they gave an explicit example of textons in this way: ''*... elongated blobs of different widths or lengths are different textons*''. In summary, textons are directly stated to be the attributes of blobs, namely length, width, orientation and colour.

An early computational implementation of texton theory was done by Voorhees and Poggio [14] where they were faithful to Julesz's textons. They proposed first-order statistics of blob attributes to determine boundaries between textures. Blob attributes were obtained on a basic multi-scale analysis on gray level images, not considering colour information. Later, Leung and Malik [1], Renninger and Malik [15] and Varma and Zisserman [16] resumed the texton theory proposing a holistic representation of textures. All of them plus Burghouts and Geusebroek [17], Burghouts and Geusebroek [18], that add colour to the description, starting from a different definition of the texton concept; they consider *textons as the cluster centres of the vectors in a filter response space*. Although these works proved to be efficient in classification tasks with specific datasets, they work with a vague definition of texton that we believe could be subtracting some of its power. We hypothesise here that the strength of a descriptor can be achieved by preserving the precision in the attribute computations.

A deep analysis on the texton concept was done by Zhu et al. [19], where textons are intuitively defined as *meaningful objects viewed at distance*, *such as stars*, *birds*, *cheetah blobs*, *snowflakes*, *beans*, *etc*. With this definition the authors propose to recover the texton shapes underlying the image generation, it is based on learning highly diverse dictionaries of texton shapes. In this work, shape of textons replaces the original concept of textons as attributes of blobs.

Recently, Liu et al. [5] also proposed to represent images through the attributes of local image regions. In this case the authors propose a simplification of image blobs by giving five special types of local templates defined as configurations of spatial 2×2 neighbourhoods, called textons, which can be seen as minimal blob regions. Co-occurrence of colour and texture features of their textors are computed and accounted in a global statistical measure (histogram) that represents the image.

Considering all previous approaches in this work we hypothesise that by going back to the original definition of texton we can provide a framework for texture description with some interesting properties:

- It provides us with a definite description of texture able to represent perceivable texture differences of any image. Texton theory is the consequence of a large experimental analysis about which texture differences are discriminated by humans and which are not.
- It is based on a computational approach that uses well-known early stages of human visual processes, namely convolution with banks of filters at different scales and the corresponding non-linear steps to select relevant information, instead of specific operators looking for all possible patterns that can be found in an image.
- It allows to integrate colour and texture in a natural manner. No special new assumptions are required, colour is added just as one more attribute of the image blobs.

In this paper, as in Voorhees and Poggio [14] and continuing the work of Alvarez et al. [20], we start with a precise definition of texton as blob attributes and we propose an image representation based on a first-order statistic of these blob attributes that can fully characterise colour-texture images. The computational representation we propose perfectly matches current bag-of-words models coming from the object recognition field [21] and also used in texture representation [15,1,17,16].

In the context of object recognition the bag-of-words (BoW) representation model has become a standard way to represent image content. Image representation is built after three main steps are done: feature detection, feature representation and vocabulary construction. In the first step significant regions of the image are extracted; in the second, features are extracted in every region. In last step the vocabulary of *visual words* is constructed by learning procedures on the feature vectors obtained from a subset of test images. Then *visual words* are the representative vectors, or prototypes, of a clustering process. Finally, the image is represented by histogram of visual words [22], without taking into account where the features are located but the frequency, in a similar way to first-order statistics of texton theory. When the location of visual words is needed as for scene categorisation problems, then hierarchical approaches are used [23], in this way a coloured texton-based hierarchical approach has been proposed in Battiato et al. [24].

One of the main problems derived from BoW approach is the way vocabularies are built and how to achieve an accurate combination of different features (e.g. texture and colour). In [25] they propose a universal vocabulary of coloured textons that is derived from combining visual words that optimise the object categorisation task. A deep discussion on how to build vocabularies that combine local shape and colour is done in [26]. Either in object categorisation Khan et al. [26] or in near-duplicate retrieval [27,28] more complex shape descriptors surpass texton concept.

In this work we propose a texture representation based on the BoW framework, representing colour-texture image content, where the features are the attributes of image blobs as stated in the texton theory. In our approach a first step is devoted to the computation of textons. To extract textons we build a multiscale Laplacian approach with further refinements to get image blobs, we call *p-blobs*, and subsequently their attributes: width, length, orientation and colour. Second, vocabulary is obtained from the direct quantisation of *p-blob* attribute spaces, without any previous learning step. In this way, we can work with low-dimensional spaces (three dimensions for shape and three for colour) providing visual words with some perceptual meaning. Thus, the first stages of our BoW framework are achieved in a pure feedforward manner with no learning. Finally, the image is represented by a probability density function of vocabulary terms (*visual words*). This representation fits the first-order statistics of textons proposed in the texton theory, where they state that pre-attentive visual system accounts for textons in the simplest global way, this is, by computing its frequency.

By building different vocabularies we can achieve different image representations. Here we explore three different models to construct vocabularies where colour and texture are combined differently. The first and most obvious vocabulary is a direct sampling on each attribute dimension, as it was done by Voorhees and Poggio [14]. This way provides a descriptor as the concatenation of all attribute frequencies (*TD* descriptor). Afterwards we improve this image description by adding spatial co-occurrence of attributes, first a full co-occurrence (JTD descriptor), second co-occurrence of colour and shape is computed separately (STD descriptor).

We evaluate the proposed descriptor on a large image dataset, composed mainly by images from the Corel collection, in different applications and we compare our performance with current state-of-art descriptors. We show that our co-occurrent descriptors outperform previous results.

The rest of the paper is organised as follows: in Section 2 we explain how we propose to build the vocabularies from a precise computation of image textons. In Section 3 we explain the basic texton descriptor. In Section 4 we take a further step over the

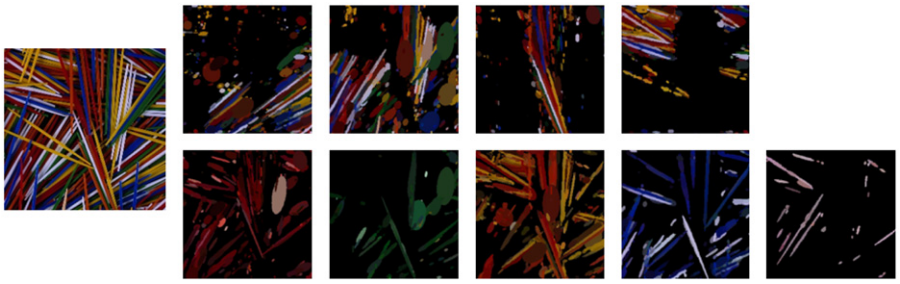

Fig. 1. Our image decomposition. Left: original image. First row: images of *p-blobs* in intervals of similar orientation (22.5°, 67.5°, 112.5°, 157.5°). Second row: images of *p-blobs* in intervals of similar colour (red, green, yellow-orange, blue, white-pink). (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this article.)

first-order statistics of Julesz and Bergen [10] by inserting co-occurrence of attributes in the vocabularies. Sections 5 and 6 explain in detail how *JTD* and *STD* descriptors are defined. Finally, in Section 7 we evaluate the proposed descriptors and we conclude with final remarks in Section 8.

2. Vocabulary of textons

As Julesz and Bergen [10], we consider textons as the attributes of line-segments and blobs. Both elements are grouped under our *p-blob* (*perceptual blob*) that refers to a region perceived as convex and with colour homogeneity.[1] In Fig. 1 we show an example of our image decomposition that allows us to build different image planes representing *p-blobs* computed and grouped accordingly with specific attributes, namely orientation in the top raw and colour in the second row.

Our texton vocabularies are constructed by direct quantisation on texton spaces. These are low-dimensional spaces of *p-blob* attributes, one for shape and one for colour. In the next sections we describe the procedure to obtain textons and afterwards we introduce the quantisation process to define the visual words and derived vocabularies. In a BoW frame, texton computation is our feature detection step, and quantisation is the vocabulary construction.

2.1. Texton computation

We work on a pure bottom-up approach to extract textons. We propose a process of five stages inspired in the works of Lindeberg [29], Lindeberg [30]. These stages are based on a linear filtering with Gaussian partial derivatives at multiple scales, and non-linear steps based on local maxima operations. This follows a pure feedforward approach in line with others low level computational models in computer vision literature [31]. The five stages needed to automatically detect *p-blobs* are summarised below.

First stage: *normalised Laplacian filtering*: In this stage we use the normalisation proposed by Lindeberg [30] for Laplacian operators in the scale-space representation. Assuming that all image blobs have Gaussian shapes, image blobs are early detected using the normalised differential Laplacian of the Gaussian operator in a subset of scales given by

$$\nabla^2_{norm} L(\cdot;\sigma) = \sigma^2 \nabla^2 L(\cdot;\sigma) \tag{1}$$

being $\sigma = \{1.284^n,\ n \in [1..11]\}$ and $L(x,y;\sigma) = I(x,y) * G(x,y;\sigma)$, where I is the original image and $G(\cdot;\sigma)$ is a Gaussian function. To avoid blob detection due to noise we apply a restriction, $\nabla^2_{norm} L(\cdot;\sigma) \geq \eta_{det}$, where η_{det} will be determined by estimating the signal–noise relationship.

[1] We introduce the *p-blob* concept to distinguish it from current uses of blob as more general low-level features.

Since blobs can emerge from intensity variations (due to surface geometry, like roughness), chromaticity variations (due to reflectance properties), or both, in order to detect image blobs, we use a colour space representation that separates chromaticity and intensity information. A basic colour space fulfilling this property is the Opponent space. Image blobs are then obtained applying the normalised Laplacian onto each opponent colour component. Previously, components are normalised to be invariant to intensity changes.

Second stage: *maximum detection over scales*: Blob centers are located where the function $\nabla^2_{norm} L$ reaches its maximum over scales, while the width of the blob (w) corresponds to the scale, s_{LoG}, of higher function value. So we compute for each detected blob its spatial localisation and its optimal scale.

Third stage: *structural tensor at integration scale*: According to Lindeberg [29] the best procedure to compute the attributes of blobs is by computing the structural tensor at the integration scale, this is done by computing the second order matrix, defined by

$$\mu_L(\cdot;t,s) = G(\cdot;s) * ((\nabla L)(\cdot;t)(\nabla L)(\cdot;t)^T) \tag{2}$$

being ∇L the image gradient evaluated at $t = s_{LoG}$ (blob scale) and G a Gaussian function with $s = \gamma s_{LoG}$, that is the integration scale of the tensor operator, and typically γ value is 2. In this way we can compute the tensor eigenvectors, (v_1,u_1) and (v_2,u_2), corresponding respectively to λ_1 y λ_2, which are the tensor eigenvalues in decreasing order.

This procedure detects a large amount of blobs of similar shapes overlapping in the space and some of them do not correspond to perceived blobs. We show an example in first row of Fig. 2 where images show the redundancy of the blobs detected in every colour channel of the left image.

Fourth stage: *local maxima over detected blobs*: This stage applies a refinement stage to remove the redundancy of the previous result and providing the *p-blobs*. To this end, a winner-take-all competition among overlapping blobs is performed. It keeps the blob of higher filter response from those overlapping with it while removing the remaining blobs. As a result of this stage in second row of Fig. 2 we show *p-blobs* obtained from previous example.

Fifth stage: *attribute computation*: Textons are finally extracted by computing all the attributes of *p-blobs* detected, these are:

- *Shape attributes*, width, length and orientation, denoted respectively as (w,l,θ) are computed using results of the second and third stages

$$w = s_{LoG},\ \theta = arctan(v_2/u_2)$$

$$l = sqrt(\lambda_1/\lambda_2) \cdot w = sqrt(\lambda_1/\lambda_2) \cdot s_{LoG} \tag{3}$$

- *Colour attributes*, are estimated using colour information from all pixels belonging to a *p-blob*. We obtain them (identified by

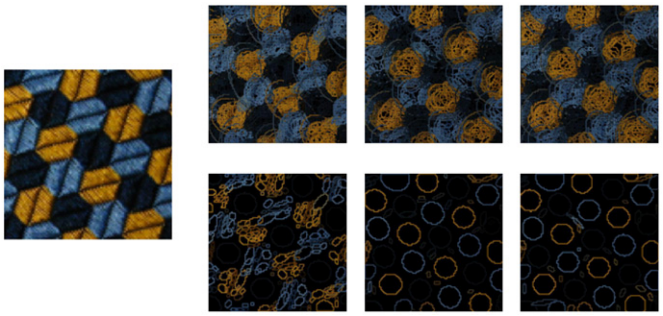

Fig. 2. Examples of perceptual blob detection. Left: original image. First row: detected blobs on *I*, *RG* and *BY* channels. Second row: *p-blobs* obtained in every colour channel respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$(\bar{i},\overline{rg},\overline{by})$) employing the median statistic operator which is robust to noise and outliers.

Thus, for a given image with h *p-blobs*, its attributes or textons are given by

$$\mathbf{W}^T=[w_1\dots w_h],\quad \mathbf{I}^T=[\bar{i}_1\dots\bar{i}_h]$$

$$\mathbf{L}^T=[l_1\dots l_h],\quad \mathbf{RG}^T=[\overline{rg}_1\dots\overline{rg}_h]$$

$$\Theta^T=[\theta_1\dots\theta_h],\quad \mathbf{BY}^T=[\overline{by}_1\dots\overline{by}_h] \tag{4}$$

2.2. Vocabulary construction

To define any vocabulary we represent *p-blob* attributes (textons) in low dimensional spaces. All attributes, either shape, (w,l,θ), and colour, (i,rg,by), convey in spaces with bounded axes which are derived either by the image dimensions or by the inherent nature of the attributes. Bounded spaces allow to build vocabularies by a direct quantisation of specific spaces. In this work, we will denote a quantisation function, $Q_{\triangle}$, as follows:

$$Q_{\triangle}:\mathbb{R}^k\rightarrow\mathbb{N}^k \tag{5}$$

where k is the dimension of the space to be quantified and subindex $\triangle$ identifies the quantisation model used.

The first and simplest vocabulary that can be built is given by the quantisation of six one-dimensional spaces, one for each attribute. We have used a quantisation model based on a sampling with bins of equal length along the one-dimensional space, this quantisation function is denoted as $Q_{\#}$, and the number of bins is denoted by m. In this way we obtain six different vocabularies, each one denoted as

$$V^X=\{x_1,\dots,x_m\} \tag{6}$$

where x_j is a *visual word* and X is the random variable that takes values in the set of different attributes, $X\in\{W,L,\Theta,I,RG,BY\}$ corresponding to the computed textons, width, length, orientation, intensity, *rg* and *by* components, respectively. Therefore the global vocabulary we propose with this first quantisation is given by

$$V=\bigcup_X V^X\quad \text{where } X\in\{W,L,\Theta,I,RG,BY\} \tag{7}$$

by quantifying each texton space in m intervals, the cardinal of the vocabulary is given by $\#V=6\times m$ terms.

3. Texton descriptor (*TD*)

Considering the vocabulary defined in the previous section now we can proceed to define the corresponding image descriptor. To this end, we need to obtain the corresponding visuals words for a given image, by applying $Q_{\#}$ function on the extracted textons (Eq. (4)), this is given by

$$\{Q_{\#}(\mathbf{W}),Q_{\#}(\mathbf{L}),Q_{\#}(\mathbf{\Theta}),Q_{\#}(\mathbf{I}),Q_{\#}(\mathbf{RG}),Q_{\#}(\mathbf{BY})\} \tag{8}$$

Considering any random variable X obtained using the quantisation function $Q_{\#}$, we define the i-component of the basic texton image representation from the probability density function of X, that is

$$P_X(x_i)=P[X=x_i] \tag{9}$$

that represents the likelihood of a particular *p-blob* attribute in the image. Then we define the basic texton image representation, *Texton Descriptor (TD)*, as the concatenation of the probability distributions of all six random variables which are related to perceptual blob attributes. This is given by

$$TD=[P_W,P_L,P_\theta,P_I,P_{RG},P_{BY}] \tag{10}$$

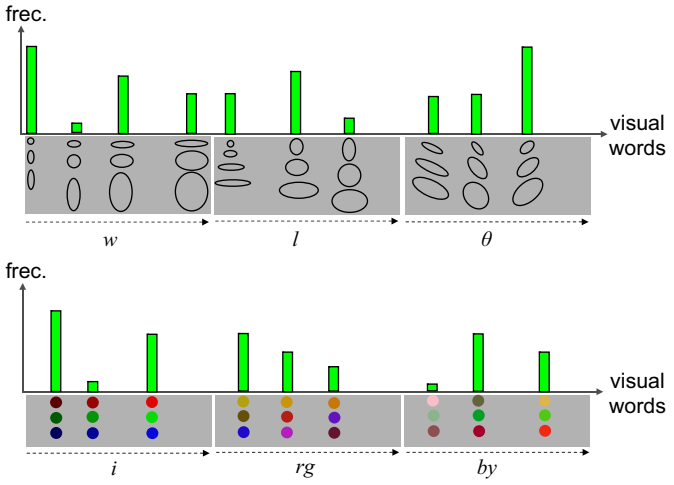

Fig. 3. Texton descriptor (*TD*).

In Fig. 3 we give an schematic example of a colour-texture representation using its Texton Descriptor. With this basic texton image representation, visual words describe colour and shape of *p-blobs*. This definition updates the basic definition of Voorhees and Poggio [32] extending it to colour. The main drawback of this descriptor lies on its inherent simplicity, co-occurrence between different textons is completely lost. One visual word can refer to different geometries or different colours. In this way, the positional relationship between textons is ignored as it was stated in texton theory. In the next sections we will explore new vocabularies defined on the textons spaces but combining texton with non-linear transforms and using different sampling functions.

4. New vocabularies of textons

The vocabulary defined in the previous section was built by a direct sampling on to the one-dimensional spaces provided by the attributes. In this way, the proposed descriptor does not exploit two useful properties that could increase its representation power, these are:

- The existence of perceptual relationships between attributes.
- The co-occurrence of shape and colour attributes at the blob level.

To introduce the perceptual relationships between attributes we propose to define new textons that are derived by non-linear transforms on the original attributes resulting in two three-dimensional

spaces with cylindrical coordinates. Then, to introduce the co-occurrence of the attributes at the blob level we propose two ways to build new vocabularies and representations.

4.1. New textons

We propose to use new textons as the result of combining the basic attributes of *p-blobs*. Combination of attributes is done by introducing uniform properties to the spaces. In colour science [33] a common approach to introduce perceptual considerations is to define uniform spaces. These spaces are built in such a way that perceptual similarities correlate with Euclidean distances.

Perceptual relationships between shape attributes are used in the shape space by a non-linear transform, denoted as $\mathcal{US}$, and defined as

$$\mathcal{US}: \mathbb{R}^3 \rightarrow \mathbb{R}^3$$
$$(w,l,\theta) \rightarrow (r,z,\phi) \tag{11}$$

where $r = \log_2(ar)$, $z = \log_2(\log_2(A))$ and $\phi = 2\theta$, being ar the blob aspect ratio ($ar = w/l$), A its area ($A = w \cdot l$) and θ its orientation. Then, two axes of this shape texton space describe the blob size and a third axis defines blob orientation. In Fig. 4 we show some blob shapes examples represented in the *shape texton space*. By definition, blob width is the lowest length of two blob lengths, then all blobs are localised inside the cone shown in Fig. 4 delimited by $\alpha_{max} = \pi/4$.

In a similar way, we apply a non-linear transform to the colour information of *p-blobs*. We use a well-know transform to HSI colour space that is also three-dimensional and uses cylindrical coordinates as well. This colour space allow us to work with perceptual axes and present similar properties to uniform colour spaces. Since our colour blob attributes have been computed in an opponent colour representation, the following transformation is needed in order to obtain new colour blob attributes:

$$\mathcal{UC}: \mathbb{R}^3 \rightarrow \mathbb{R}^3$$
$$(i,rg,by) \rightarrow (h,s,i) \tag{12}$$

thus, new colour blob attributes are (h,s,i), corresponding to the perceptual properties of hue, saturation and intensity respectively.

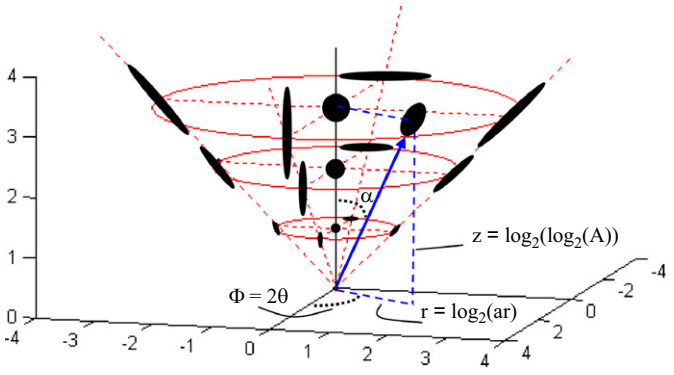

Fig. 4. Shape texton space in cylindrical coordinates.

4.2. Vocabulary construction

Once different perceptual texton spaces have been built, vocabularies can be constructed using different quantisation models. In this section we explore different possibilities in order to evaluate how they can better represent colour-texture properties.

Texton spaces are bounded and their axes have perceptual properties, these two interesting properties allow a generation of vocabularies by a direct quantisation of these spaces. The obtained vocabularies are universal since they do not depend on any specific training set of textures and their sizes are determined by the number of bins used in the quantisation process.

We have explored three different quantisation models on texton spaces to build different vocabularies. We refer to them as Cartesian, cylindrical and circular. The three models have been used for the shape texton spaces and only the first two have been used for colour. They are explained in more detail below:

- Cartesian model, $M_{\#}$, sampling process is shown in Fig. 5(a). Spaces are uniformly quantified using the same number of bins in each dimension. In colour spaces this has been previously used by Lee et al. [34], others have used a finer quantisation on chromatic axes and a coarser quantisation on the achromatic axis as Swain and Ballard [35].
- Cylindrical model, $M_{\otimes}$, bins are shown in Fig. 5(b). This model exploits the benefits of having perceptual axes in texton spaces, therefore the quantisation can be directly applied on each texton independently, these are (r,z,ϕ) and (h,s,i) for shape and colour respectively.
- Circular model, $M_{\circledcirc}$, again sampled bins are shown in Fig. 5(c). This model is a variant of the previous one but improving the quantisation on the central area. This case is specially useful for shape space where isotropic blobs are located near the vertical axis. In this way they are clearly separated from non-isotropic blobs.

For the case of shape texton space we define the quantisation function as $Q^S_{\triangle}$, (Eq. (5)) being $\triangle \in \{\#, \otimes, \circledcirc\}$ with $k=3$. For each one of the models we denote m_1, m_2 and m_3 as the number of bins for the three axes respectively. We obtained a vocabulary, V^S, that only describe shape of *p-blobs* and which cardinality is given by $\#V^S = 1/2 \times (m_1 \times m_2 \times m_3)$ visual words.

For the colour texton space we define the quantisation function as $Q^C_{\triangle}$, being $\triangle \in \{\#, \otimes\}$ again with $k=3$. To represent colour

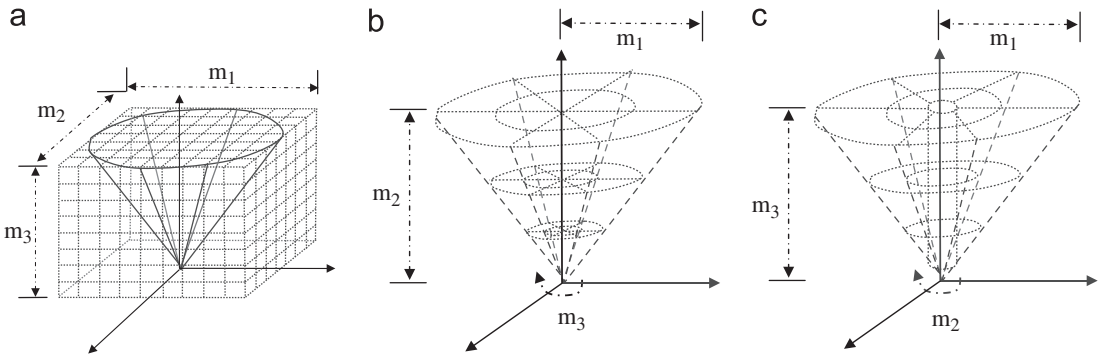

Fig. 5. Different quantisation models. (a) Model $M_{\#}$, (b) Model $M_{\otimes}$ and (c) Model $M_{\circledcirc}$.

information of *p-blobs* we have explored two different spaces of the HSI family, these are *HSI-Carron* [36] and *HSV-Smith* [37] spaces, distribution of colours in these spaces is shown in Fig. 6. Using models $M_{\#}$ and $M_{\otimes}$ texton spaces have been quantified with n_1, n_2 and n_3 number of bins per axis. The resulting vocabulary, V^C, which only describes colour of *p-blobs*, presents cardinality $\#V^C = n_1 \times n_2 \times n_3$ visual words.

Previous vocabularies are separately defined for shape and colour. The combination of colour and shape attributes can be done by spatial co-occurrence of both attributes at the blob level or separately at the image representation level. Depending on the combination two different vocabularies are constructed:

Co-joint vocabulary is based on the assumption that coloured texture are characterised by shapes of specific colours. We need a vocabulary where a visual word jointly describes these two properties. Therefore we construct the vocabulary V^{JCS} where each visual word represents the co-occurrence of shape and colour attributes. This vocabulary can be built by a six-dimensional combination of attributes from both spaces and is formed by $\#V^{JCS} = \#V^C \times \#V^S = (n_1 \times n_2 \times n_3) \times (1/2 \times m_1 \times m_2 \times m_3)$ terms.

Semi-joint vocabulary is based on the assumption that coloured texture can be characterised by shape and colour attributes separately; in this way visual words are considered without any relationship between them and therefore the vocabulary, V^{SCS}, is built by a direct union of previous vocabularies, this is: $V^{SCS} = V^S \cup V^C$, which cardinality is given by $\#V^{SCS} = \#V^C + \#V^S = (n_1 \times n_2 \times n_3) + (1/2 \times m_1 \times m_2 \times m_3)$.

5. Co-joint texton descriptor (*JTD*)

This descriptor represents the image content using the vocabulary V^{JCS}. In this way the *Co-joint Texton Descriptor (JTD)* is defined as the probability density function of a bi-dimensional random variable (C,S), which is composed of two discrete random variables: C that belongs to the quantised colour texton space and S that belongs to the quantised shape texton space

$$JTD = [P_{C,S}(c_1,s_1), \ldots, P_{C,S}(c_M,s_N)] \quad (13)$$

where $P_{C,S}(c_i,s_j) = P[C = c_i, S = s_j]$ is the joint probability density function of colour and shape attributes. Then the *JTD* descriptor introduces full co-occurrence of colour and shape attributes to represent colour-textures. As a consequence, the proposed vocabulary

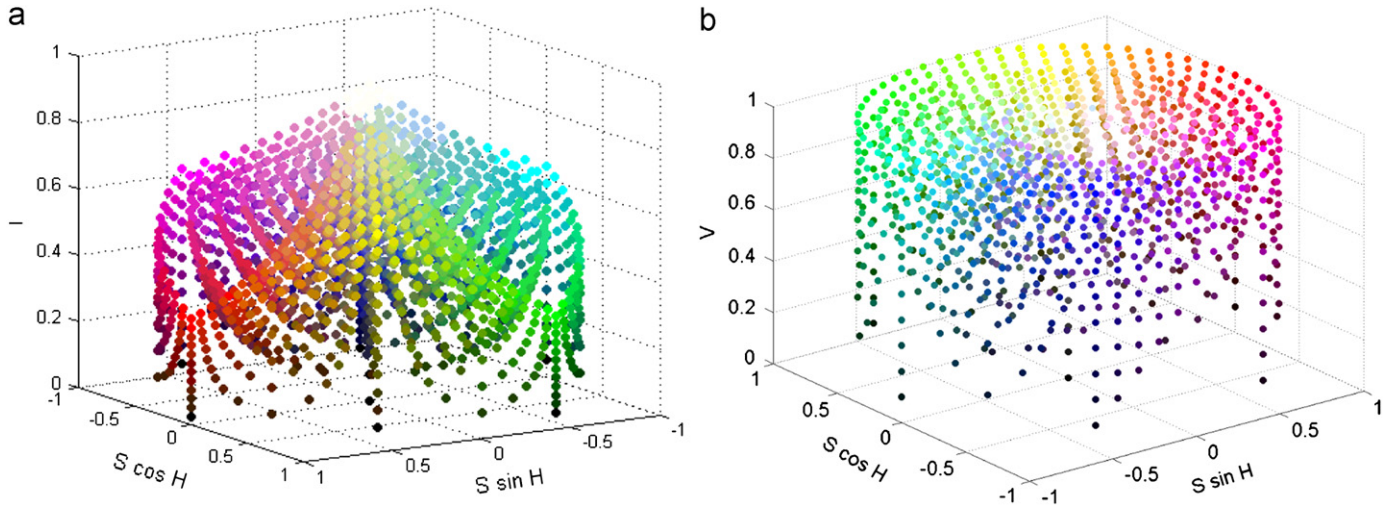

Fig. 6. Colour spaces used to represent *p-blob* colour information. (a) *HSI-Carron colour space*, (b) *HSV-Smith* colour space. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this article.)

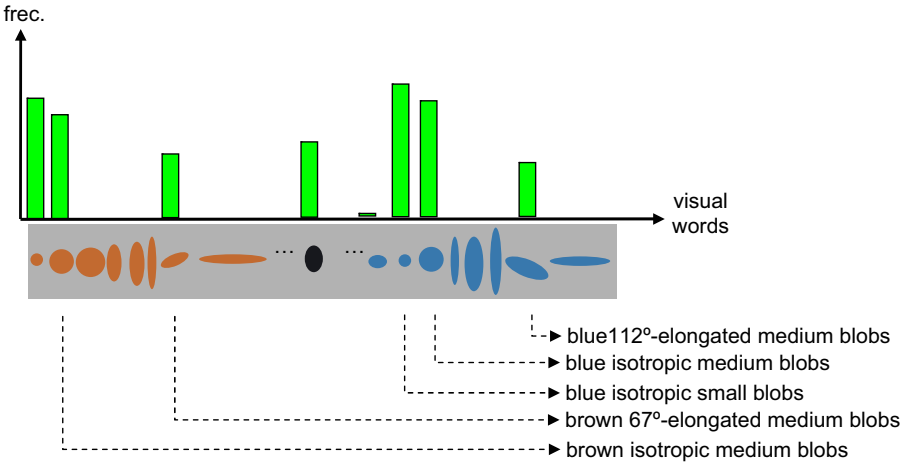

Fig. 7. Co-joint texton descriptor (*JTD*) in a one-dimensional representation.

has a clear correspondence between a visual word and all the attributes co-occurring on a *p-blob*.

Fig. 7 shows the *JTD* descriptor of the image in Fig. 2, where its perceptual visual words is graphically shown. We have used linguistic terms (adjectives) to refer to the attributes associated to each blob. Also in Fig. 8 we show some examples of this image representation.

6. Semi-joint texton descriptor (*STD*)

In this section we propose a second colour-texture representation that uses a vocabulary where, again, colour and texture are combined. The vocabulary is V^{SCS}, where words describe shape and colour of *p-blobs*. The *PTD* is similar to the *JTD* but removes the co-occurrence between colour and shape attributes.

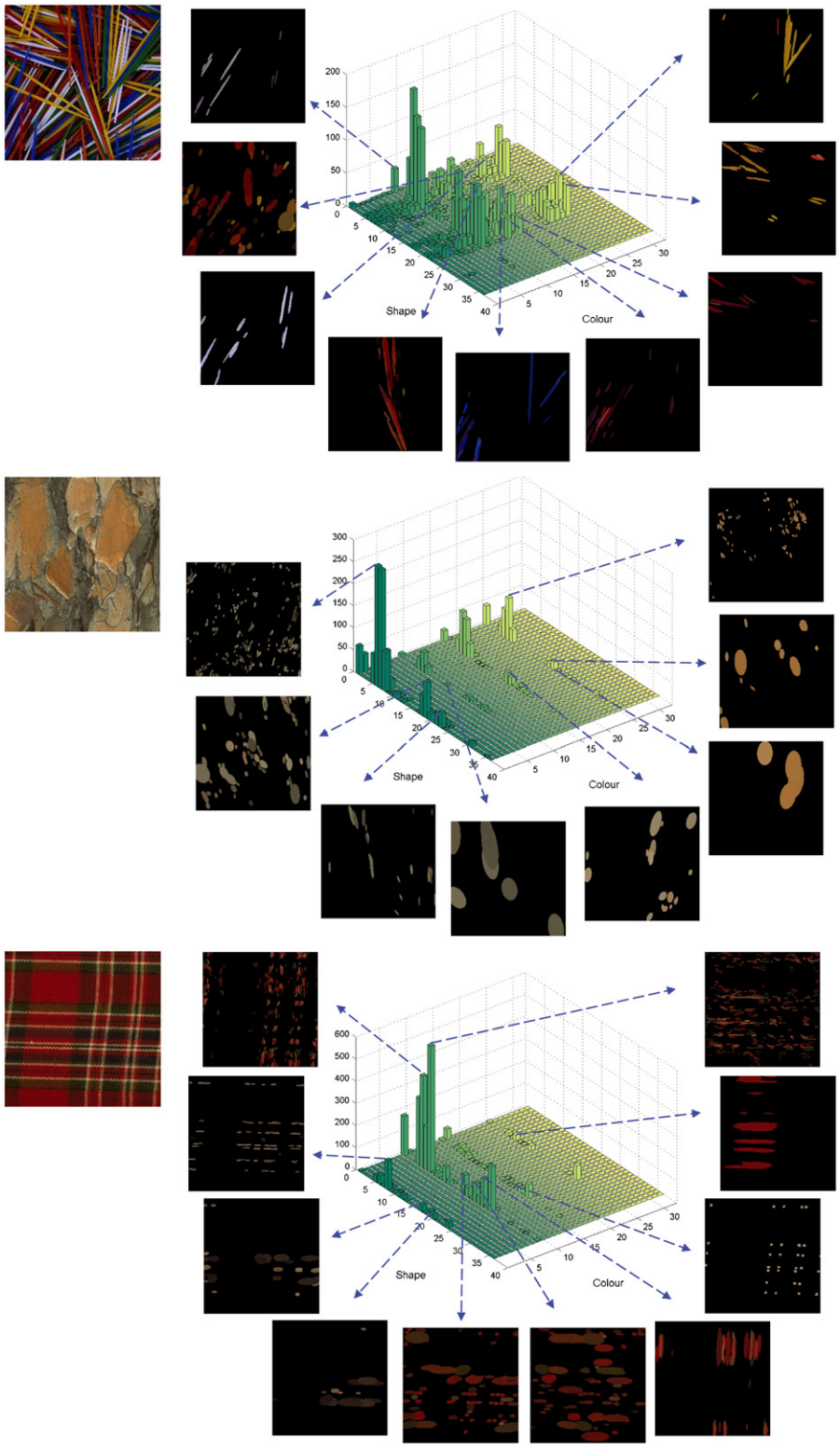

Fig. 8. At left some textures examples and at right their *JTD* representation including its correspondent *p-blob* decomposition.

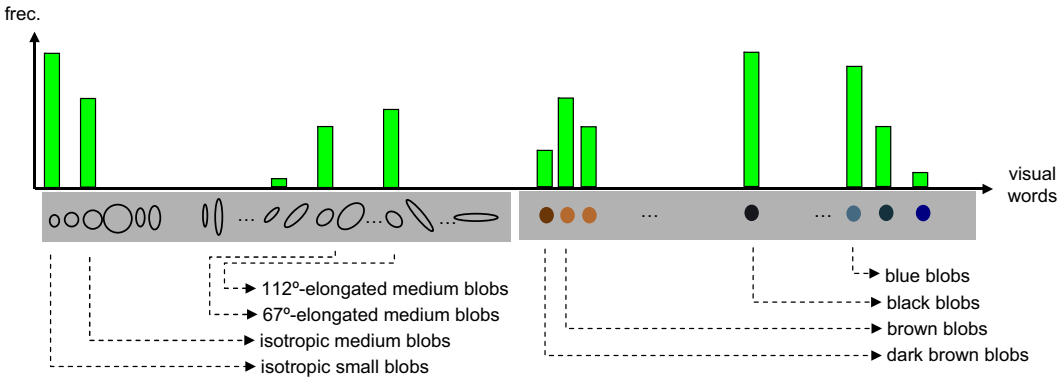

Fig. 9. Semi-joint texton descriptor (*STD*) in a one-dimensional representation.

We define the *Semi-joint Texton Descriptor* (*STD*) from marginal probabilities of bidimensional random variable (*C*,*S*) as follows:

$$STD = [STD_s\ STD_c]$$

$$STD_s = [P_S(s_1), \ldots, P_S(s_N)]$$

$$STD_c = [P_C(c_1), \ldots, P_C(c_M)] \quad (14)$$

where $P_C(c) = \sum_{s_j} P_{C,S}(c_i,s_j) \equiv \sum_{s_j} P[C = c_i, S = s_j]$ is the marginal probability of random variable *C*, and $P_S(s) = \sum_{c_i} P_{C,S}(c_i,s_j) \equiv \sum_{c_i} P[C = c_i, S = s_j]$ is the marginal probability of random variable *S*.

Fig. 9 shows an example of a *STD* descriptor, where again, for each perceptual visual word we display its associated linguistic terms.

7. Experimental results

Two different applications have been used to evaluate the performance of proposed colour-texture representations: image retrieval and classification. In image retrieval the goal is to find images similar to a query image, while in image classification the aim is to identify the class of the query image from pre-trained classes. Both applications need to define relevant sets, either a composition of classes in classification or a set of similar images in retrieval. To construct relevant sets we follow a common setup: given an image we built its class or obtain its similar images dividing every image in *J* non-overlaping sub-images.

7.1. Datasets

Here we list diverse datasets we have used in the experiments:

Corel, all datasets from the Corel stock photography collection[2] related to colour-textures. Below we indicate its names and in brackets its corresponding references and nicknames: Textures (137,000, *CorelTex*), Textures II (404,000, *CorelTex2*), Various Textures I (593,000, *CorelV1Tex*), Various Textures II (594,000, *CorelV2-Tex*), Textile Patterns (192,000, *CorelTexPat*), Sand & Pebble Textures (390,000, *CorelSand*), Bark Textures (399,000, *CorelBark*), Colours & Textures (403,000, *CorelCol*), Marble Textures (349,000, *CorelMarb*), Painted Textures (265,000, *CorelPain*), Shell Textures (355,000, *CorelShel*). Each Corel group has 100 textures and each relevant set has $J=6$ sub-images. Then, the total number of textures is $6 \times 100 = 600$ for each Corel dataset. In Fig. 10 we show some samples of each dataset.

Outex, dataset (*TC-00013*) [38], it has 68 textures and $J=20$ sub-images per texture.

Vistex,[3] we consider two similar subsets: the dataset used by Mäenpää and Pietikäinen [3] with 54 textures, that we call *VisTexP*, and the dataset defined by Liapis and Tziritas [39] with 55 textures, here identified as *VisTexL*. Both datsets have $J=16$ sub-images.

7.2. Distance metric

We have used the chi-square (χ^2) metric to compare our image descriptors since they are probability distributions. Given two image descriptors, H_1 and H_2, their similarity measure is computed by the χ^2 distance as follows:

$$\chi^2(H_1,H_2) = \frac{1}{2}\sum_{t=1}^{T}\frac{[H_1(t)-H_2(t)]^2}{H_1(t)+H_2(t)} \quad (15)$$

7.3. Image retrieval

We carried out a series of experiments using different vocabularies obtained by varying the feature quantisation model, and in the case of the *JTD* and the *STD* descriptors, also varying the colour space to represent colour information.

To assess quantitatively the performance of the retrieval we have used two standard measures *Recall* [40] and *Precision* vs. *Recall* graphs. The retrieval *Recall* is defined as follows:

$$recall(n) = \frac{N_{relevant}}{relevant} \quad (16)$$

where $N_{relevant}$ is the number of relevant images in the *n* images retrieved and *relevant* is the total number of relevant images in the database. The results have been computed by using all images of each dataset as query images, then we compute the *average Recall* as a percentage:

$$\overline{recall}(n) = \frac{1}{P}\sum_{i=1}^{P} recall_i(n) \times 100 \quad (17)$$

where $recall_i$ is the *Recall* of image *i* and *P* is the number of images in the database. In the ideal case of the retrieval, if $n=relevant$ then the *average Recall* will be 100. The retrieval *precision* is

[2] Corel data are distributed through http://www.emsps.com/photocd/corelcds.htm.

[3] Vision Texture-VisTex database (MIT Media Lab) http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html.

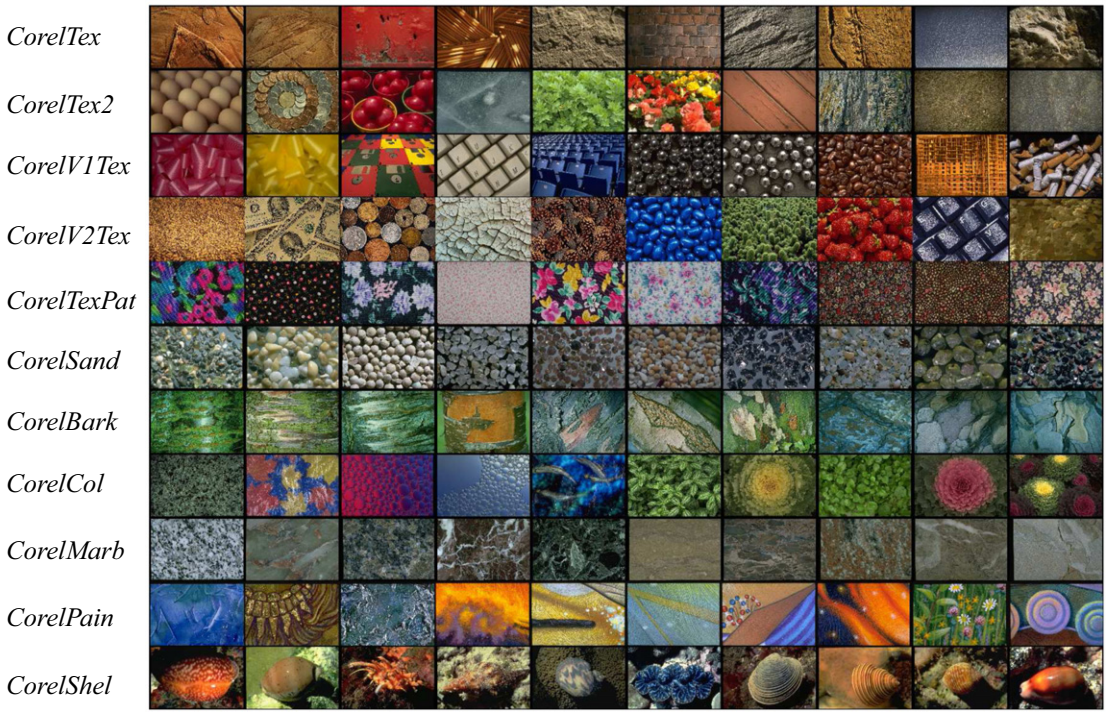

Fig. 10. In each row some examples of each one of the 11 colour-texture Corel databases used in experimentation. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this article.)

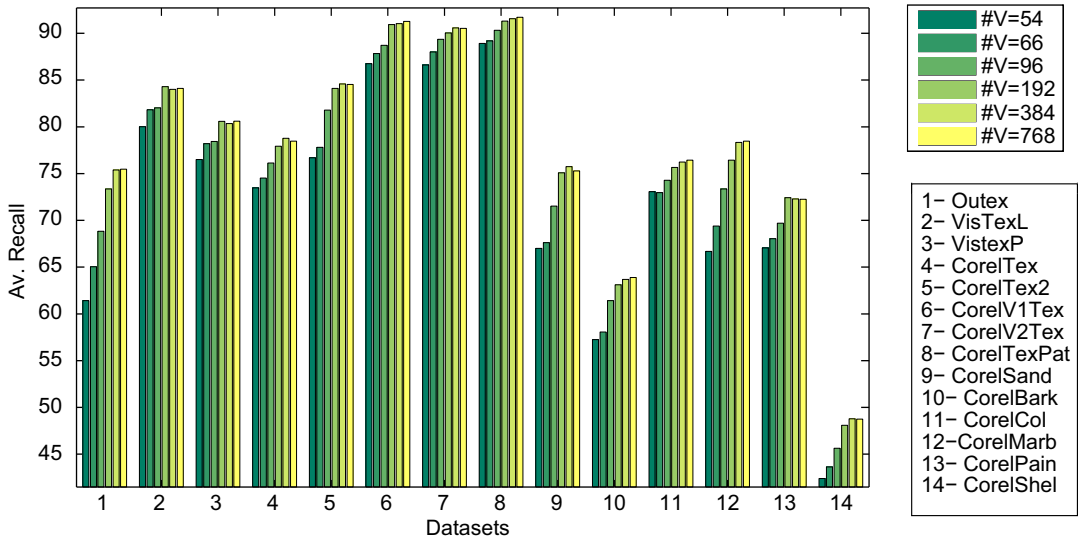

Fig. 11. *Average recall* of *TD* for each dataset using different vocabularies.

defined as follows:

$$precision(n) = \frac{N_{relevant}}{n} \tag{18}$$

where n is the total number of returned images by the retrieval.

7.3.1. Experiment 1: TD evaluation

In this experiment, the colour-texture representation is evaluated using the descriptor *TD*. We have tested six vocabularies quantifying respectively each texton in $m = 9, 11, 16, 32, 64$ and 128 intervals. The *average Recall* obtained in each dataset is shown in Fig. 11 where we detail the results using the mentioned vocabularies of size $\#V = 6 \times m$ terms.

From the results obtained we can state that vocabulary size determines the efficiency of the *TD* descriptor: increasing the vocabulary size also increases its efficiency; however this is not a critical parameter. Vocabulary size cannot be too small in order to avoid that the same word refers to different textons and hence has to have a sufficient number of terms; this is more than 96 terms. From this number on the efficiency of the descriptor has little variation.

For comparison purposes we have determined the best vocabulary, therefore, for each vocabulary; we have computed the mean of *average Recall* obtained in all datasets displayed in Fig. 11; in Table 1 we show the mean values. The highest mean has been achieved with vocabularies of 384 and 768 terms. From these two, the smallest vocabulary will be used in a later section.

7.3.2. Experiment 2: JTD evaluation

Different vocabularies has been obtained using the mentioned feature quantisation models and varying quantisation parameters (number of bins). Also, we have used two different colour spaces to represent colour information, *HSI-Carron* and *HSV-Smith*. For each vocabulary we have done the same experiment obtaining the *Recall* measure in order to evaluate all the parameters involved in the quantisation process that derives different vocabularies. To easily identify each test we have coded every experiment; Table 2 shows codes and parameters used in the construction of the vocabularies. For each vocabulary we detail the quantisation model, the number of bins in quantising texton spaces and vocabulary size. For each quantisation model a greater number of encoding corresponds to a greater number of intervals in quantisation process and therefore a higher vocabulary size.

Fig. 12(a) shows the *average Recall* percentages obtained for each dataset using the *HSV-Smith* colour space to represent colour information in all tested vocabularies. In this graph and for every database, vocabularies giving a highest *Recall* value were highlighted with bigger points. The *Recall* obtained varies widely depending on dataset; this is due in part to the non-homogeneity of relevant sets in *Outex* and *CorelShel* databases. The best vocabularies are $q9$, $q13$ and $q16$, corresponding to models where colour texton space has been quantised in a circular way ($Q^C_{\otimes}$). To assess the average behavior of the *JTD* descriptor we have computed the mean *Recall* of all datasets that we show in Fig. 12(b). In the same figure we also show the results of the same experiment but using *HSI-Carron* colour space to represent colour information. This graph shows that the best vocabulary is $q13$ where shape texton space has also been quantised, as colour texton space, in a circular way ($Q^S_{\otimes}$) this vocabulary will be used for comparing purposes in a later section. Another conclusion is that in the different quantisation models we have tested the behavior of *HSV-Smith* colour space is better than *HSI-Carron*.

Table 1
Mean *Recall* of all datasets with *TD* descriptor using different vocabularies.

#V	54	66	96	192	384	768
Mean	73.18	74.47	76.36	78.46	**79.02**	78.95

7.3.3. Experiment 3: STD evaluation

In order to evaluate the *STD* descriptor we have repeated the same experiment performed in previous descriptor, using different vocabularies varying quantisation parameters and using two different colour spaces to represent colour information (*HSI-Carron* and *HSV-Smith*). Again, for simplification purposes, we have coded every experiment, as shown in Table 3. Indices of the

Table 2
Codification of tested vocabularies in *JTD* evaluation.

Quantisation models	Vocabulary size					(m_1,m_2,m_3), $\#V^{JCS}$ (n_1,n_2,n_3)
$Q^S_{\#}$, $Q^C_{\#}$	(5,5,5), 7812 (5,5,5) q_1	(5,5,5), 30 375 (9,9,6) q_2	(6,6,6), 27 648 (8,8,4) q_3	(7,7,7), 43 904 (8,8,4) q_4	(7,7,7), 83 349 (9,9,6) q_5	(5,5,7), 52 500 (10,10,6) q_6
$Q^S_{\#}$, $Q^C_{\otimes}$	(5,5,5), 16 000 (4,4,16) q_7	(7,7,7), 43 904 (4,4,16) q_8	(5,5,7), 50 400 (6,6,16) q_9			
$Q^S_{\otimes}$, $Q^C_{\otimes}$	(4,5,8), 10 240 (4,4,8) q_{10}	(4,5,8), 20 480 (4,4,16) q_{11}	(4,7,8), 28 672 (4,4,16) q_{12}	(3,7,8), 48 384 (6,6,16) q_{13}		
$Q^S_{\otimes}$ $Q^C_{\otimes}$	(3,7,8), 21 504 (4,4,16) q_{14}	(3,7,8), 33 600 (5,5,16) q_{15}	(3,7,8), 48 384 (6,6,16) q_{16}			

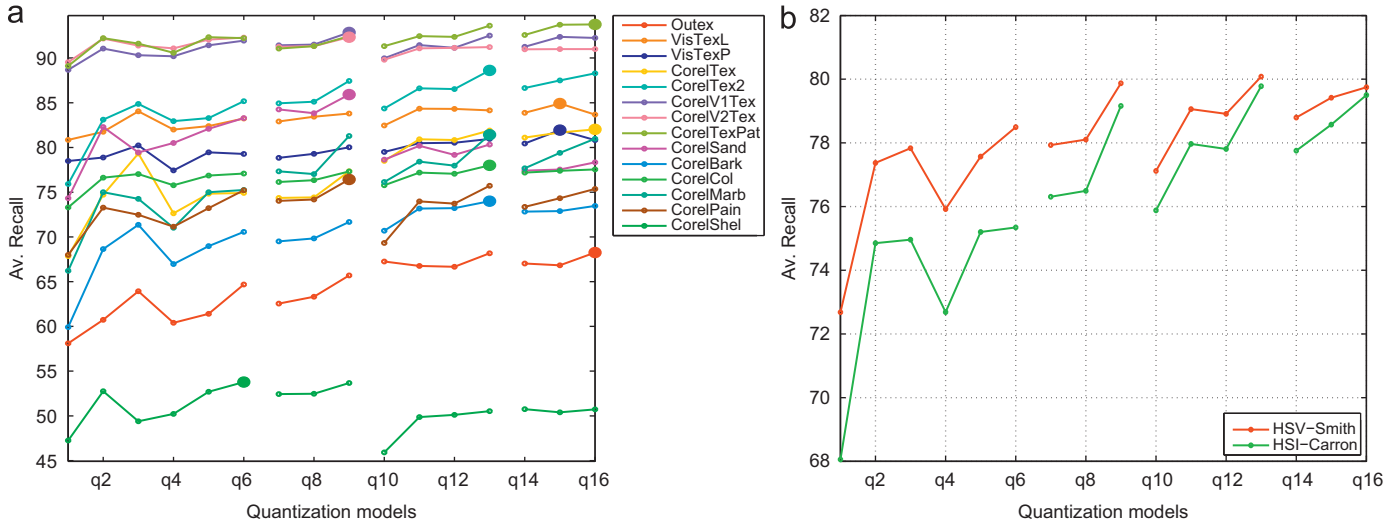

Fig. 12. *JTD* descriptor evaluation (a) with different quantisation models for each dataset and (b) mean *Recall* of all datasets.

Table 3
Codification of tested vocabularies in *STD* evaluation.

Quantisation models	Vocabulary size					(m_1,m_2,m_3), $\#V^{SCS}$ (n_1,n_2,n_3)
$Q^S_{\#}$, $Q^C_{\#}$	(5,5,5), 549 (9,9,6) q_1	(7,7,7), 658 (9,9,6) q_2	(9,9,9), 1094 (9,9,9) q_3	(6,6,6), 1404 (12,12,9) q_4		
$Q^S_{\#}$, $Q^C_{\otimes}$	(5,5,5), 319 (4,4,16) q_5	(7,7,7), 1468 (9,9,16) q_6	(9,9,9), 1660 (9,9,16) q_7			
$Q^S_{\otimes}$, $Q^C_{\otimes}$	(4,5,8), 208 (4,4,8) q_8	(4,5,8), 336 (4,4,16) q_9	(4,7,8), 368 (4,4,16) q_{10}	(4,7,8), 1488 (9,9,16) q_{11}		
$Q^S_{\circledcirc}$ $Q^C_{\otimes}$	(3,7,8), 340 (4,4,16) q_{12}	(3,7,8), 660 (6,6,16) q_{13}	(3,7,8), 1108 (8,8,16) q_{14}	(3,7,8), 1380 (9,9,16) q_{15}	(4,7,8), 1408 (9,9,16) q_{16}	(3,7,8), 1684 (10,10,16) q_{17}

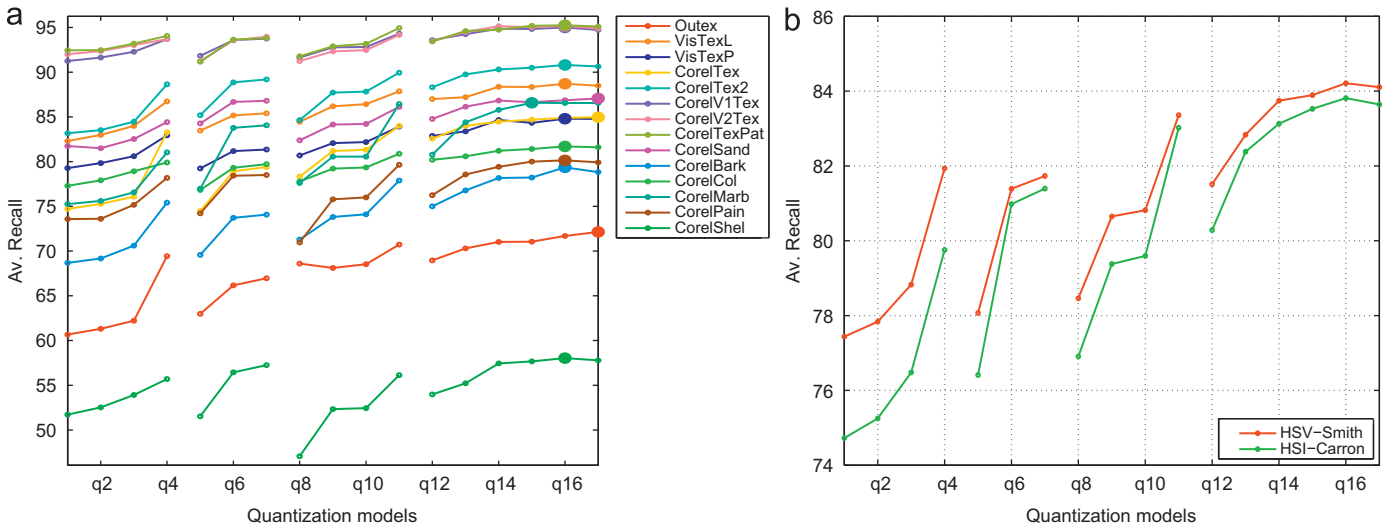

Fig. 13. *STD* descriptor evaluation (a) with different quantisation models for each dataset and (b) mean *Recall* of all datasets.

code also correlate with a greater number of intervals in quantisation process and therefore with higher vocabulary size.

Fig. 13(a) shows the *average Recall* percentages for each dataset using different vocabularies and *HSV-Smith* colour space to represent colour information. In regard to the *JTD* descriptor, a similar variation of *Recall* measure among datasets is also observed in this graph. However, in this case, one of the vocabularies emerge as the best, $q16$, where colour and shape spaces are quantised in a circular way. In particular, shape texton space has been quantised using the special quantisation model $M_{\circledcirc}$ that exploit better the features of shape texton space. This conclusion is also obtained from the information shown in Fig. 13(b), where we show the mean *Recall* of all datasets and again the behavior of *HSV-Smith* colour space is better than *HSI-Carron*.

7.3.4. Comparison with other descriptors

In order to compare our results we have used the two image representation models mentioned in the introduction, that integrates colour and texture in different ways. In the model where colour and texture are processed separately we have used *MPEG-7* descriptors [2] and *MTH* descriptor [5]. In the model where colour and texture are jointly processed we have used the *LBP RGB* descriptor [3].

Among *MPEG-7* descriptors two of them have been used to describe colour information, these are *SCD* and *CSD* descriptors. The first one describes colour distributions while the second describes colour organisation inside images. To describe texture information the *HTD* descriptor is useful for homogeneous textures and the *EHD* for non-homogeneous textures. *MPEG-7* is a multimedia content description standard where there is not a defined procedure to combine descriptors, for this reason we have adopted the method of Dorairaj and Namuduri [7] that combines dissimilarity measures.

The *MTH* descriptor combines colour and texture orientation in a texton histogram and has been used recently in an image retrieval application over large Corel datasets.

In case of *LBP* (*RGB*) descriptor, it has some parameters and can be itself combined [3,41,42], for this reason three different models have been tested, these are: $LBP_{8,1}RGB$, $LBP(^{u2}_{8,1}+^{u2}_{16,3}+^{u2}_{24,5})RGB$ and $LBP(^{u2}_{8,1}+^{u2}_{16,2}+^{u2}_{24,3})RGB$ referenced in this paper as LBP_1, LBP_2, LBP_3 respectively.

We have done the same retrieval experiment over all datasets using *MTH* descriptor and the above-mentioned combination for *MPEG-7* and *LBP* descriptors. Table 5 shows the *average Recall* obtained in each dataset for all descriptors. The highest *Recall* is indicated in boldface and in cursive the second best *Recall* to highlight the best results for each dataset.

In the first three columns of Table 5 comparing only our descriptors, we show our best performance achieved using *TD* ($\#V=384$), *JTD* (q13) and *STD* (q16). Results in that table show

that *STD* descriptor is superior in all datasets, except for *Outex* where *TD* performs best and *VistexL* for which is reported a performance of 91.3% by Liapis and Tziritas [39]. For all datasets, *STD* has a superior *average Recall* than *JTD* at the expense of a smaller vocabulary size (the *STD* descriptor has 1408 terms and the *JTD* descriptor 48,384). Comparing all descriptors, *STD* outperforms all the others in almost all datasets and it has the best behaviour on average. The second best descriptor is *JTD* followed by a combination of MPEG-7 descriptors (*EHD+CSD*).

To compare the performance of previous descriptors in a higher dataset we have done an additional experiment: we have joined all Corel images in a single dataset, thus decreasing inter-class variability. The results of the *average Recall* obtained appear in Table 4. Performance analysis of proposed descriptors has also been evaluated using *Precision* vs. *Recall* graphs presented in Fig. 14. Both results proved that the behaviour of descriptor *STD* is clearly better than *LBP*, *MTH* and *MPEG-7* descriptors, while performance of *JTD* and *EHD+CSD* is very similar.

All this experimentation carried out on a very diverse datasets has demonstrated the usefulness of the descriptor *STD* to model any kind of colour-textures and its superior performance compared to other well known descriptors.

Table 4
Recall of the Corel datasets union.

Descriptor	*MTH*	LBP_3	*EHD+CSD*	*TD*	*JTD*	*STD*
Recall	40.11	60.01	75.80	45.30	75.90	**82.43**

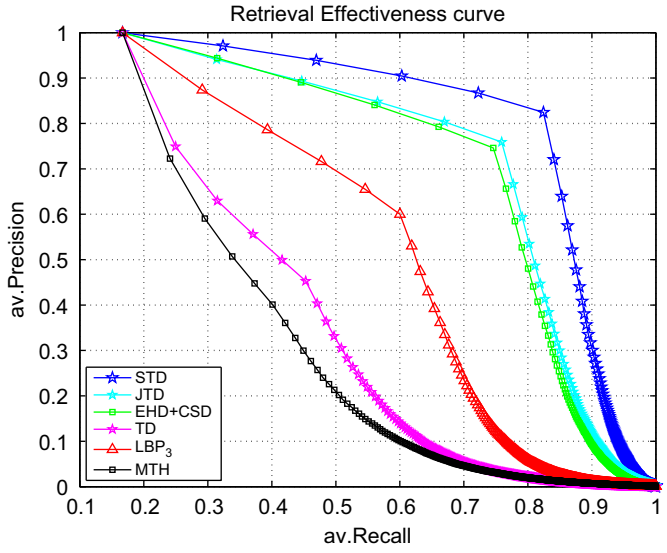

Fig. 14. *Precision* vs. *Recall* graph of Corel datasets union.

7.3.5. Texture and colour contribution on STD

Further experimentation has been done over *STD* in order to study the contribution of shape and colour components of this descriptor separately. We have repeated the retrieval experiment, however, we have only used the component of *STD* that characterises blob shape distribution on colour-textures, this is STD_s (Eq. (14)) and used the component that characterise blob colour distribution, STD_c. To construct the vocabulary of each component we have used the same parameters of quantisation model *q16* and the *HSV-Smith* colour space, obtaining 1296 terms for STD_c and 112 terms for STD_s. Table 6 lists the results of this experiment in contrast of the *STD* results, where the highest *Recall* of both components is indicated in boldface.

We can state that STD_c component always achieves better performance compared with STD_s component in all datasets. In some of them (*CorelSand*, *CorelMarb*, *CorelPain* and *CorelV1Tex*) the contribution of STD_c is much higher than STD_s contribution, thus indicating greater discrimination power of colour descriptor. This experiment also show that the performance of the *STD* descriptor is better than their components by themselves.

Table 6
Average Recall of *STD* components.

BD	STD_s	STD_c	*STD*
Outex	48.18	**61.83**	71.69
VisTexL	63.49	**81.60**	88.70
VisTexP	62.37	**78.35**	84.80
CorelTex	56.78	**75.86**	84.89
CorelTex2	59.58	**86.61**	90.81
CorelV1Tex	55.72	**92.17**	95.00
CorelV2Tex	67.39	**91.19**	95.17
CorelTexPat	73.89	**92.53**	95.25
CorelSand	36.50	**85.11**	86.86
CorelBark	46.33	**72.33**	79.36
CorelCol	58.04	**74.87**	81.70
CorelMarb	44.72	**81.97**	86.56
CorelPain	40.30	**77.03**	80.14
CorelShel	32.00	**53.94**	58.03
Mean	53.24	**78.96**	84.21

Table 5
Average *Recall*.

BD	*STD*	*JTD*	*TD*	*HTD+SCD*	*HTD+CSD*	*EHD+CSD*	LBP_1	LBP_2	LBP_3	*MTH*
Outex	*71.69*	67.17	**75.39**	61.10	66.02	67.64	56.87	46.82	60.80	39.99
VisTexL	**88.70**	84.15	84.01	*87.41*	83.45	85.25	73.20	50.36	66.11	61.44
VisTexP	**84.80**	80.82	80.35	*83.44*	80.24	81.81	70.38	45.85	64.69	57.43
CorelTex	**84.89**	*81.89*	78.78	67.33	72.36	75.31	61.89	60.83	62.22	41.61
CorelTex2	**90.81**	*88.61*	84.58	76.11	85.89	87.33	72.50	68.89	72.42	55.47
CorelV1Tex	**95.00**	92.50	91.03	85.94	91.89	*92.78*	77.53	72.14	74.56	68.00
CorelV2Tex	**95.17**	91.22	90.58	88.53	85.89	*92.28*	81.47	79.81	81.44	69.08
CorelTexPat	**95.25**	*93.61*	91.56	93.69	89.86	90.50	86.89	82.92	86.03	58.78
CorelSand	**86.86**	80.33	75.75	72.00	85.06	*85.28*	55.54	60.17	59.89	45.78
CorelBark	**79.36**	*74.00*	63.69	64.83	69.11	70.36	48.39	51.69	51.67	42.06
CorelCol	**81.70**	*78.01*	76.24	73.43	72.78	75.52	68.89	68.73	70.62	53.24
CorelMarb	**86.56**	81.42	78.33	70.67	80.11	*82.03*	59.78	63.72	65.19	42.36
CorelPain	**80.14**	*75.72*	72.28	62.31	72.61	73.94	50.67	49.11	51.69	52.78
CorelShel	**58.03**	*50.53*	48.78	41.50	47.06	49.97	32.83	35.94	35.67	30.83
Mean	**84.21**	*80.00*	77.95	73.45	77.31	79.29	64.06	59.78	64.46	51.36

Table 7
Classification rates.

BD	*STD*	*JTD*	*HTD+SCD*	*HTD+CSD*	*EHD+CSD*	LBP_1	LBP_3	*MM*	*MTH*	*MR8-LF*
Outex	*90.32*	84.71	75.78	86.71	90.04	71.51	76.07	**94.1**	66.79	63.85
VisTexL	*99.25*	97.35	95.69	**99.56**	99.13	94.12	87.67	–	85.93	78.02
VisTexP	**98.89**	98.52	94.66	*98.53*	98.24	92.11	95.95	97.9	83.02	75.02
CorelTex	**93.73**	*92.02*	65.45	80.90	86.88	73.78	75.85	–	43.72	55.08
CorelTex2	**97.78**	*96.88*	75.38	88.97	95.38	84.57	83.32	-	64.07	75.33
CorelV1Tex	**97.63**	96.13	84.04	94.15	*96.83*	88.12	87.15	–	78.48	80.05
CorelV2Tex	**98.77**	96.08	86.45	95.07	*96.98*	90.90	90.57	–	78.30	86.43
CorelTexPat	**98.48**	*97.42*	94.98	97.87	95.28	96.35	94.47	–	66.82	84.12
CorelSand	**92.83**	80.88	63.32	78.13	*92.10*	61.05	67.42	–	49.10	60.08
CorelBark	**91.10**	*85.08*	56.58	70.57	84.95	55.78	64.17	–	43.37	48.23
CorelCol	**89.37**	*86.35*	73.81	86.12	85.57	82.43	81.72	–	58.80	64.27
CorelMarb	**95.45**	*90.38*	65.70	80.78	91.73	70.53	76.33	–	41.77	58.18
CorelPain	**97.98**	*86.03*	54.42	72.00	85.97	57.13	59.77	–	57.75	48.17
CorelShel	**72.32**	59.62	32.82	38.47	*61.02*	31.68	32.27	–	25.50	28.50
Mean	**93.85**	89.10	72.79	83.42	*90.01*	75.00	76.62	–	60.24	64.67

7.4. Image classification

Image classification is one of the most frequently used application to test descriptors viability. Image classification requires two steps: the first is learning, where a subset of images belonging to each relevant set is needed for training the classifier and, in the second step the rest of the relevant images are used to test the classifier. We used the simplest classifier, nearest neighborhood, because the goal is to test the descriptor and not to obtain the best classification rate.

In this experiment half of the images of each dataset have been used for training and the rest for test. The classification process has been repeated 20 times using in each learning step a different random set of images. Table 7 presents the average classification rates of our best descriptors (*STD*(q13) and *JTD*(q16)), *LBP*, the best combination of MPEG-7 descriptors and *MTH* descriptor. In this classification experiment we also have added a comparison with *MR8-LF* descriptor, based on the texton model of Varma and Zisserman [16] that uses the MR8 filter bank; it has been extended to colour following the procedure suggested by Burghouts and Geusebroek [17] (vocabulary is independently built for each colour channel, we learn 10 words per channel, 30 words per training image and we learn from 20 images randomly drawn from 20 different classes, obtaining 600 words). In the same table we have also included *MM* descriptor proposed by Arvis et al. [43], where the authors reported results on two of the datasets we used, achieving the best score in one of them. For each dataset the highest rate is indicated in boldface and the second best rate in italics.

Results in this experiment show that the *STD* descriptor outperforms remaining descriptors in 12 of 14 datasets, and in the two datasets where is not the best descriptor it achieves the second best rate. On an average the *STD* descriptor outperforms in 3.84% next best rate achieved by *EHD+CSD* descriptors that has a similar performance to our descriptor *JTD*. These results confirm conclusions obtained in previous experiments and demonstrates the good behavior of our descriptors also in texture characterisation.

8. Conclusions

In this paper we revisit texton theory [10] presenting a new computational approach that is faithful to the original definition of textons, defined as the attributes of image blobs. We use a refined procedure to extract blobs and compute their shape attributes, size, length, orientation and colour.

We propose several descriptors in the BoW framework, thus the density of features perfectly match first-order statistics of the texton theory. Our descriptors are built by direct quantisation on the spaces of blob attributes without needing any learning stage. In this way our proposal is taking the advantage of the optimised representations used by perceptual systems for texture discrimination instead of training from specific image datasets. Quantisation of these texton spaces provides universal texture vocabularies, which visual words have a direct translation to linguistic terms.

Differences between proposed descriptors rely on how attributes are combined. The *TD* descriptor concatenates individual attribute distributions at a late step (no co-occurrence of attributes); on the other hand, the six blob attributes are fused at an early step in the *JTD* descriptor (full blob co-occurrence). Finally, the *STD* descriptor concatenates shape and colour blob attributes (colour blob co-occurrence separated from shape blob co-occurrence). The experiments carried out with different and diverse image datasets have shown an efficient performance of our descriptors in representing coloured texture images. We report an extensive evaluation and comparison of our descriptors showing important improvements on current state-of-art in image retrieval and classification applications. These results bring us to the conclusion that the *STD* descriptor shows a slightly better performance in most experiments. This may mean that avoiding co-occurrence of colour and shape features at the blob levels maintains a high-level of average performance with a small vocabulary size. However, the *JTD* descriptor with a larger vocabulary achieves an important discriminatory power for specific image subsets.

Further research on these low-level descriptors can be directed to insert perceptual grouping mechanisms capturing the properties of spatial patterns than can emerge from blobs configurations. This could be integrated as further higher-level features added to the proposed framework.

Appendix A. Supplementary data

Supplementary data associated with this article can be found in the online version at http://dx.doi.org.10.1016/j.patcog.2012.04.032.

References

[1] T. Leung, J. Malik, Representing and recognizing the visual appearance of materials using three-dimensional textons, International Journal of Computer Vision 43 (2001) 29–44.
[2] B. Manjunath, J. Ohm, V. Vinod, A. Yamada, Color and texture descriptors, IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on MPEG-7 11 (2001) 703–715.

[3] T. Mäenpää, M. Pietikäinen, Classification with color and texture: jointly or separately? Pattern Recognition 37 (2004) 1629–1640.
[4] D. Lowe, Distinctive images features from scale-invariant keypoints, International Journal of Computer Vision 60 (2004) 91–110.
[5] G.-H. Liu, L. Zhang, Y.-K. Hou, Z. yong Li, J.-Y. Yang, Image retrieval based on multi-texton histogram, Pattern Recognition 43 (2010) 2380–2389.
[6] Y. Chun, N. Kim, I. Jang, Content-based image retrieval using multiresolution color and texture features, IEEE Transactions on Multimedia 10 (2008) 1073–1084.
[7] R. Dorairaj, K. Namuduri, Compact combination of MPEG-7 color and texture descriptors for image retrieval, in: Conference of the 37th Asilomar on Signals, Systems and Computers, vol. 1, 2004, pp. 387–391.
[8] H. Yu, M. Li, H. Zhang, J. Feng, Color texture moments for content-based image retrieval, in: International Conference on Image Processing, 2003, pp. 24–28.
[9] Y. Zhong, A.K. Jain, Object localization using color, texture and shape, Pattern Recognition 33 (2000) 671–684.
[10] B. Julesz, J. Bergen, Textons the fundamental elements in preattentive vision and perception of textures, Bell Systems Technological Journal 62 (1983) 1619–1645.
[11] J. Bergen, Theories of visual texture perception, in: D. Regan (Ed.), Vision and Visual Dysfunction, vol. 10B, 1991, pp. 114–134.
[12] B. Julesz, E. Gilbert, J. Victor, Visual discrimination of textures with identical third-order statistics, Biological Cybernetics (1978) 137–140.
[13] A. Rao, G. Lohse, Towards a texture naming system: identifying relevant dimensions of texture, Vision Research 36 (1996) 1649–1669.
[14] H. Voorhees, T. Poggio, Computing texture boundaries from images, Nature 333 (1988) 364–367.
[15] L. Renninger, J. Malik, When is scene recognition just texture recognition? Vision Research 44 (2004) 2301–2311.
[16] M. Varma, A. Zisserman, A statistical approach to texture classification from single images, International Journal of Computer Vision 62 (2005) 61–81.
[17] G. Burghouts, J. Geusebroek, Material-specific adaptation of color invariant features, Pattern Recognition Letters 30 (2009) 306–313.
[18] G. Burghouts, J. Geusebroek, Color textons for texture recognition, in: Proceedings of the British Machine Vision Conference, vol. 3, 2006, pp. 1099–1108.
[19] S.-C. Zhu, C.-E. Guo, Y. Wang, Z. Xu, What are textons? International Journal of Computer Vision 62 (2005) 121–143.
[20] S. Alvarez, A. Salvatella, M. Vanrell, X. Otazu, Perceptual color texture codebooks for retrieving in highly texture datasets, in: International Conference on Pattern Recognition, 2010, pp. 866–869.
[21] J. Sivic, A. Zisserman, Video google: a text retrieval approach to object matching in videos, in: IEEE Conference on Computer Vision and Pattern Recognition, 2003, pp. 1470–1477.
[22] L. Fei-Fei, P. Perona, A bayesian hierarchical model for learning natural scene categories, in: International Conference on Computer Vision and Pattern Recognition, vol. 2, 2005, pp. 524–531.
[23] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: Conference on Computer Vision and Pattern Recognition, 2006, pp. 2169–2178.
[24] S. Battiato, G.M. Farinella, G. Gallo, D. Ravì, Scene categorization using bags of textons on spatial hierarchy, in: Proceedings of the IEEE International Conference on Image Processing 2008, pp. 2536–2539, http://dx.doi.org/10.1109/ICIP.2008.4712310.
[25] J. Winn, A. Criminisi, T. Minka, Object categorization by learned universal visual dictionary, IEEE International Conference on Computer Vision 2 (2005) 1800–1807.
[26] F.S. Khan, J. van de Weijer, M. Vanrell, Modulating shape features by color attention for object recognition, International Journal of Computer Vision 98 (2011) 49–64.
[27] Y. Hu, X. Cheng, L.-T. Chia, X. Xie, D. Rajan, A.-H. Tan, Coherent phrase model for efficient image near-duplicate retrieval, IEEE Transactions on Multimedia 11 (2009) 1434–1445.
[28] S. Battiato, G. Farinella, G.C. Guarnera, T. Meccio, G. Puglisi, D. Ravì, R. Rizzo, Bags of phrases with codebooks alignment for near duplicate image detection, in: Proceedings of the ACM Workshop on Multimedia in Forensics, Security and Intelligence, 2010, pp. 65–70.
[29] T. Lindeberg, Discrete derivative approximations with scale-space properties: a basis for low-level feature detection, Journal of Mathematical Imaging and Vision 3 (1993) 349–376.
[30] T. Lindeberg, Scale-space theory: a basic tool for analysing structures at different scales, Journal of Applied Statistics 21 (1994) 225–270.
[31] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, T. Poggio, Robust object recognition with cortex-like mechanisms, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (2007) 411–426.
[32] H. Voorhees, T. Poggio, Detecting textons and texture boundaries in natural images, in: IEEE Conference on Computer Vision, 1987, pp. 250–258.
[33] G. Wyszecki, W. Stiles, Color Science—Concepts and Methods, Quantitative Data and Formulae, John Wiley & Sons, 1982.
[34] S. Lee, J. Xin, S. Wesland, Evaluation of image similarity by histogram intersection, Color Research and Applications 30 (2005) 265–274.
[35] M. Swain, D. Ballard, Colour indexing, International Journal of Computer Vision 7 (1991) 11–32.
[36] T. Carron, P. Lambert, Color edge detector using jointly hue, saturation and intensity, in: International Conference on Image Processing, vol. 3, 1994, pp. 977–981.
[37] A. Smith, Color gamut transform pairs, SIGGRAPH Computer Graphics 12 (1978) 12–19.
[38] T. Ojala, T. Mäenpää, M. Pietikäinen, J. Viertola, J. Kyllönen, S. Huovinen, Outex - new framework for empirical evaluation of texture analysis algorithms, in: Proceedings of the International Conference on Pattern Recognition, vol. 1, 2002, pp. 701–706.
[39] S. Liapis, G. Tziritas, Color and texture image retrieval using chromaticity histograms and wavelet frames, IEEE Transactions on Multimedia 6 (2004) 676–686.
[40] J. Smith, Image retrieval evaluation, in: Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries, 1998, pp. 112–113.
[41] T. Mäenpää, M. Pietikäinen, J. Viertola, Separating color and pattern information for color texture discrimination, in: Proceedings of the International Conference on Pattern Recognition, vol. 1, 2002, pp. 668–671.
[42] V. Takala, T. Ahonen, M. Pietikäinen, Block-based methods for image retrieval using local binary patterns, in: Proceedings of the Scandinavian Conference on Image Analysis, 2005, pp. 882–891.
[43] V. Arvis, C. Debain, M. Berducat, A. Benassi, Generalization of the cooccurence matrix for colour images: application to colour texture classification, Image Analysis Stereology 23 (2004) 63–72.

Susana Álvarez, was born in La Coruña (Spain), 10 de December de 1966. She received computer science engineering from the Universidad Politécnica de Cataluña (UPC, Barcelona) in 1992. Since then, she has been active in computer vision research and has been specialist in texture analysis and processing. She received her PhD from the Universitat Autònoma de Barcelona (UAB) in 2010. Since 1992 has been active in seven national research projects and has published several papers inside the scientific area. She is currently an associate professor of computer science at the Universitat Rovira i Virgili (Tarragona, Catalonia, Spain).