Int J Comput Vis (2010) 88: 447–460
DOI 10.1007/s11263-009-0315-0

Using Basic Image Features for Texture Classification

M. Crosier · L.D. Griffin

Published online: 13 January 2010

Abstract Representing texture images statistically as histograms over a discrete vocabulary of local features has proven widely effective for texture classification tasks. Images are described locally by vectors of, for example, responses to some filter bank; and a visual vocabulary is defined as a partition of this descriptor-response space, typically based on clustering. In this paper, we investigate the performance of an approach which represents textures as histograms over a visual vocabulary which is defined geometrically, based on the Basic Image Features of Griffin and Lillholm (Proc. SPIE 6492(09):1–11, 2007), rather than by clustering. BIFs provide a natural mathematical quantisation of a filter-response space into qualitatively distinct types of local image structure. We also extend our approach to deal with intra-class variations in scale. Our algorithm is simple: there is no need for a pre-training step to learn a visual dictionary, as in methods based on clustering, and no tuning of parameters is required to deal with different datasets. We have tested our implementation on three popular and challenging texture datasets and find that it produces consistently good classification results on each, including what we believe to be the best reported for the KTH-TIPS and equal best reported for the UIUCTex databases.

Keywords Texture classification · Basic Image Features · Textons

M. Crosier (✉) · L.D. Griffin
Computer Science, University College London, Gower Street, London WC1E 6BT, UK
e-mail: m.crosier@cs.ucl.ac.uk

L.D. Griffin
e-mail: l.griffin@cs.ucl.ac.uk

1 Introduction

Effective general-purpose analysis of texture in images is an important step towards a variety of computer vision applications, from industrial inspection to scene and object recognition. Its challenge lies in the wide variety of possible textures—ranging in nature from regular to stochastic and in origin from albedo variations to 3D surface structure—and the conditions under which they are imaged. Changes in lighting geometry and intensity or in camera viewpoint can have a significant impact on appearance (Leung and Malik 2001).

Any texture analysis relies on an appropriate *representation*, and the task which has become canonical as a test of representation is multi-class classification.

One paradigm which has proved effective for coping with the problems described above is to represent texture images statistically as histograms over a discrete vocabulary of local features (Leung and Malik 2001; Cula and Dana 2001; Varma and Zisserman 2005, 2009; Hayman et al. 2004; Lazebnik et al. 2003; Zhang et al. 2006; Varma and Garg 2007; Ojala et al. 2002). Images are probed locally by considering, for example, the responses to a filter bank or the greyscale values of a local image patch. These descriptor responses are then assigned to discrete bins according to some partition of the feature space.

This model encompasses two approaches to image representation. In the first (Varma and Zisserman 2005, 2009; Hayman et al. 2004; Varma and Garg 2007; Ojala et al. 2002), every image in the dataset is represented as a histogram over a common dictionary and some form of histogram comparison measure is used to compare images. This dictionary is most often defined by a once-and-for-all clustering of feature vectors from a subset of images from the dataset, as described below. The second approach

(Lazebnik et al. 2003; Zhang et al. 2006) uses a separate dictionary for each image and represents the image as a 'signature': a table of feature definitions (e.g. cluster centres) with the corresponding numbers of occurrences in the image. Image signatures are compared using a measure such as the Earth Mover's Distance. This dictionary is most often defined by clustering feature vectors from the single image to be represented.

Various classification schemes have been explored for both of these approaches, from nearest-neighbour matching (Varma and Zisserman 2005, 2009; Lazebnik et al. 2003) to kernel-based SVMs (Zhang et al. 2006; Hayman et al. 2004). Although the superiority of SVMs for texture classification has been clearly demonstrated (Caputo et al. 2005; Hayman et al. 2004; Zhang et al. 2006), nearest-neighbour is still often used as an uncommitted mechanism to compare texture representations due to its simplicity and absence of parameters that need to be tuned.

Of these three dimensions of statistical texture representation—the choice of histogram or signature representation; the descriptive space over which the histogram bins are defined; and the actual choice of histogram bins—the first two have been well-studied. The relative merits of histogram- and signature-based approaches are explored in tandem with classification schemes, and a variety of local descriptors have been proposed including:

- The joint responses of various filter banks (Varma and Zisserman 2005; Hayman et al. 2004; Leung and Malik 2001; Cula and Dana 2001), made up of e.g. Gaussian derivative filters (Hayman et al. 2004; Varma and Zisserman 2005).
- Grey-scale image patches (Varma and Zisserman 2009) or points sampled in some regular local configuration (Ojala et al. 2002); and the related notion of Markov Random Fields (Varma and Zisserman 2009).
- Modified SIFT (Lowe 1999) and intensity domain SPIN images (Lazebnik et al. 2003; Zhang et al. 2006).
- Local fractal dimension and length (Varma and Garg 2007).

In this paper we are interested in the third dimension: how to choose a dictionary of discrete features over which an image can be represented. For the sake of clarity, this paper uses the language of the common dictionary/histogram approach to representation, although many points are also relevant to signature-based approaches.

1.1 Partitioning Feature Space

The simplest way to partition feature space in order to allow a histogram representation of texture would be by regular (lattice-like) binning. However, as the dimensionality of the space 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, raising the risk of overfitting the data.

Konishi and Yuille (2000) worked around this problem by limiting the number of filters used and adaptively calculating bin widths for each dimension based on data from the training set. However, Varma and Zisserman (2004) showed that it is possible to achieve classification results approaching the state-of-the-art even with relatively high-dimensional histograms of equally-spaced bins (their best result using 5 bins in each of 8 dimensions, for 200^2-pixel images), noting that most bins remained empty. Figure 8 of Varma and Zisserman (2004) shows how the optimal number of bins represents a trade-off between the imprecision of overly-crude binning and the problems of overfitting to noisy data when the representation is too high-dimensional. The requirement for a representation based on equally-spaced binning to fall between these extremes therefore makes explicit an upper limit on the dimensionality of the feature space.

The partitioning scheme which has come to dominate is comparatively low-dimensional and controls the number of bins in the representation by defining a partition of feature space through unsupervised clustering of feature vectors into *textons* (Hayman et al. 2004; Varma and Zisserman 2005, 2009; Leung and Malik 2001; Cula and Dana 2001; Varma and Garg 2007). Local descriptors calculated from a number of training images for a given texture class are used to populate a feature space which is partitioned into a pre-selected number (typically 10–40) of regions, each represented by a cluster-centre. This is repeated for each texture class in the dataset and the combined list of cluster-centres (containing perhaps 250 to 2500 elements, depending on the clustering parameters and number of texture classes) used to Voronoi partition feature space, by labelling new descriptor vectors according to the nearest cluster-centre in feature space.

Varma and Zisserman (2005) investigated reducing redundancy in this representation by combining textons whose cluster-centres fall close to each other in feature space. This produces a slight degradation of classification performance, as does learning textons from only a subset (around half) of the total classes in a dataset.

This unsupervised clustering step almost universally employs the k-means algorithm. Jurie and Triggs (2005) noted that k-means produces poor dictionaries of features for describing natural images (for which similar descriptions to those used for texture have been studied) because of the highly nonuniform distribution of descriptor responses. This results in most k-means cluster-centres being concentrated in high-density regions of feature space, with Voronoi cells radiating outwards, so that the assignment of labels to potentially informative mid-frequency (of occurrence) descriptor responses is dominated by less informative (and poten-

tially noisy) high-frequency responses. Although this non-uniformity is less severe for texture images (which may be one of the reasons why unadorned bag-of-words representations have proved more successful in this domain), the problems with k-means still apply—including the question of how to choose a suitable value of k. Jurie and Triggs (2005) compare k-means with an acceptance-radius based clusterer for visual dictionary generation and demonstrate significant improvements in object classification results from the latter.

There are other more general problems with schemes which use unsupervised clustering to generate a feature dictionary. The need to populate feature space sufficiently to allow clustering still imposes restrictions on the choice of local description, although this can be ameliorated by sampling descriptions from a greater number of training images. More problematic is the cost of performing a nearest neighbour computation to assign each new descriptor response—at every point in an image—to a texton.

1.2 Keypoint Detection as Feature Space Quantisation

Specifying the quantisation of feature space used to define a visual dictionary can also be seen as encompassing the choice of how to sample features from an image, which is often described as an additional dimension of statistical texture representation.

An image representation histogram can be populated from the image either densely (considering every point), or from keypoints only (e.g. in Lazebnik et al. 2003; Zhang et al. 2006). Detectors used to select these keypoints are generally tuned to local aspects of the image different than the descriptors. A dual way of contrasting these two approaches is as alternative partitions of some feature space. Consider a feature space consisting of the joint response of (i) the descriptor and (ii) the information used in the keypoint detector, e.g. in the case of the Harris corner detector, x- and y-derivatives at each point in a local window (Harris and Stephens 1988). Then, in the same way that methods which describe an image densely correspond to a dense (generally Voronoi) partitioning of feature space, those using keypoint detection assign labels only to those points which fall within an appropriate sub-region of feature space as determined by the rules of the keypoint detector, ignoring the remainder, i.e. a non-dense partition is induced. That is, detecting keypoints in an image can be seen as equivalent to performing some form of implicit feature selection in this joint response space.

1.3 A Geometrically Defined Partition of Feature Space

In this paper, we investigate the classification performance of an approach which represents textures as histograms over a feature dictionary which is defined mathematically—by the type of local geometry—rather than by clustering.

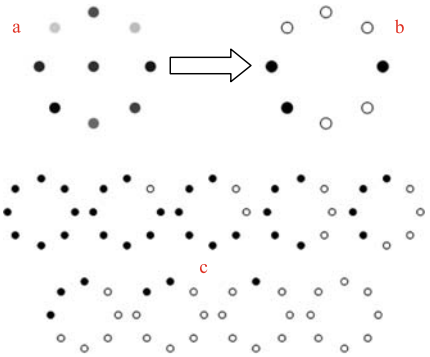


Fig. 1 Local binary patterns. (**a**) Sampling points from the image, with $P = 8, R = 1$. (**b**) Binarisation to get $\mathrm{LBP}_{8,1}$. (**c**) The set of uniform patterns $\mathrm{LBP}^{riu2}_{8,1}$

We describe an image locally at some scale using a family of six Gaussian derivative filters and base our visual dictionary on the partition of this response space defined by the Basic Image Features of Griffin and Lillholm (2007). The idea is to assign each filter response vector to one of a set of Basic Image Features (BIFs), each corresponding to a qualitatively different type of local geometric structure, based on a study of types of local symmetry (see Sect. 2). In our current scheme there are seven such BIFs which are calculated mathematically by deciding which of seven simple combinations of filter response values is largest.

As well as avoiding the problems inherent in using k-means clustering, our approach has the advantages over clustering methods of simplicity—there is no need for a pre-training step to learn a visual dictionary—and computational efficiency, since we assign filter responses to histogram bins without needing to perform a nearest-neighbour computation.

1.4 Related Work

Statistical texture representations which are based on equally-spaced binning of feature space or on visual dictionaries derived by clustering feature vectors are discussed above.

One approach which, like ours, engineers a dataset-independent dictionary of local features over which textures are represented statistically, is Local Binary Patterns (LBPs) (Ojala et al. 2002). Images are probed locally by sampling greyscale values at a point g_c and P points $g_0, \ldots, g_{P-1}$ spaced equidistantly around a circle of radius R (the choice of which acts as a surrogate for controlling the scale of description) centred at g_c, as shown in Fig. 1a. The resulting feature space of $P + 1$ greyscale values can be partitioned according to one of a nested set of progressively more invariant LBP systems:

– The first defines Local Binary Patterns themselves. The greyscale value at g_c is subtracted from those at $g_0, \ldots,$

g_{P-1} and the resulting values thresholded about zero to produce a Local Binary Pattern (as in Fig. 1b), $\mathrm{LBP}_{P,R}$, given by $\mathrm{sign}[g_0 - g_c], \ldots, \mathrm{sign}[g_{P-1} - g_c]$, which is by definition invariant to any monotonic greyscale transformation.

- Rotation invariance is built in by factoring out cyclic relabelling of $g_0, \ldots, g_{P-1}$, i.e. representing each group of LBPs which are equal under some cyclic relabelling of $g_0, \ldots, g_{P-1}$ by a single canonical LBP (denoted $\mathrm{LBP}^{ri}_{P,R}$).
- Since the dimensionality of the representation (which grows exponentially with P) is still high, a form of feature selection based on complexity is employed. Uniform LBPs ($\mathrm{LBP}^{riu2}_{P,R}$) are those (rotationally invariant) patterns which contain at most two transitions between 0 and 1, as shown in Fig. 1c. In many cases, the majority of patterns observed in texture images are classified as one of these $P + 1$ uniform LBPs. All other LBPs are grouped together into a single 'other' category, producing a $P + 2$ dimensional representation.

LBPs are similar to our approach in that they are based upon a pre-defined visual dictionary rather than one derived with reference to the dataset to be analysed. They therefore share the advantages listed above over methods based on clustering. They also possess similar invariances to our method. The central difference results from the local description used: we probe an image locally using Gaussian derivative filters where as LBPs sample greyscale values. This allows us to make use of some powerful mathematical properties of Gaussian derivatives in order to study the local geometry of the image in a way that allows a more geometrically rigorous treatment of invariances and partitioning of feature space. For example, the steerability (Freeman and Adelson 1991) of Gaussian derivative filters allows us to achieve exact rotation invariance rather than the approximate rotation invariance of LBPs.

The remainder of the paper is structured as follows: In Sect. 2 we introduce Basic Image Features and our BIF-based texture representation. In Sect. 3 we evaluate this approach against a selection of state-of-the-art alternatives on a commonly used texture dataset. In Sect. 4 we extend our approach to incorporate scale invariance: this involves extending our representation and developing a multi-scale texture comparison metric for classification. Results are presented on two additional datasets which contain significant intra-class changes in scale.

2 Basic Image Features (BIFs)

Basic Image Features (Griffin and Lillholm 2007, 2008; Griffin 2007, 2008; Griffin et al. 2009) are defined by a partition of the filter-response space (*jet space*) of a set of six

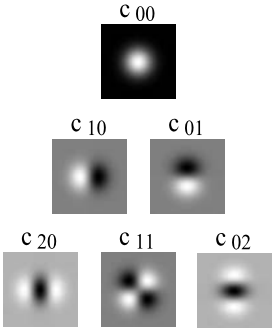


Fig. 2 Our filter bank, consisting of one zeroth-order, two first-order and three second-order Gaussian derivative filters, all at the same scale. We refer to the vector of responses as a *local jet*

Gaussian derivative filters (Fig. 2). These filters provide an uncommitted front-end to describe an image locality fully up to second order at some scale. For example, the steerability of Gaussian derivatives (Freeman and Adelson 1991) means that the response to any rotated first or second order filter can be calculated as a linear combination of the (first or second order resp.) partial derivative filters of Fig. 2.

Jet space is partitioned into seven regions—which we refer to as BIFs—each corresponding to one of seven qualitatively distinct types of local image structure, based on symmetry types (Fig. 3). Algorithm 1 defines this partition by assigning a given filter response vector to one of the seven BIFs. An example of an image 'labelled' with BIFs in this way is given in Fig. 3.

Algorithm 1 *Calculation of BIFs. The single parameter ε controls what amplitude of structure is tolerated before a region is no longer considered sufficiently uniform to be assigned to the 'flat' (pink) BIF category (see Fig. 3), and is given another label.*

1. *Measure filter responses c_{ij}, and from these calculate the scale-normalised filter responses $s_{ij} = \sigma^{i+j} c_{ij}$*
2. *Compute $\lambda = s_{20} + s_{02}$, $\gamma = \sqrt{(s_{20} - s_{02})^2 + 4s_{11}^2}$*
3. *Classify according to the largest of:* $\{\varepsilon s_{00}, 2\sqrt{s_{10}^2 + s_{01}^2}, \pm\lambda, 2^{-\frac{1}{2}}(\gamma \pm \lambda), \gamma\}$

There are two stages to the derivation of this partition. In the first, information about the local structure of the scene (*intrinsic* information) is separated from information resulting from uninteresting changes in imaging setup (*extrinsic* information). In the second, this intrinsic component is quantised into regions corresponding to different types of local image symmetries.

The transformations which are considered uninteresting for the purpose of calculating BIFs are rotations, reflections, intensity multiplications and addition of a constant intensity. Jet space is factored (Griffin 2007) by these extrinsic transformation groups to produce an intrinsic component in which all filter responses differing only in one of these extrinsic factors are mapped to the same point. Any partition of

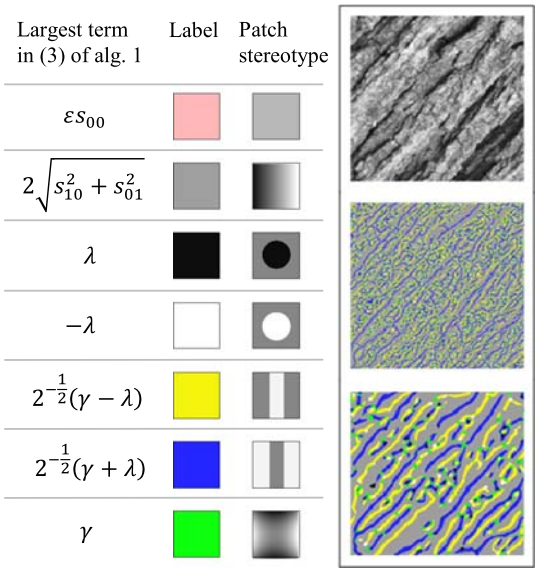


Fig. 3 *Left*: Key to the labelling used, with stereotypical image patches demonstrating the type of structure/symmetry represented by each of the seven BIFs defined by step 3 of Algorithm 1. *Right*: An image of bark from the UIUCTex database (Lazebnik et al. 2003), densely labelled with BIFs computed at scales $\sigma = 1$ and $\sigma = 4$ (both with $\varepsilon = 0$), according to the colours of the key, in order to show where different BIFs occur in a real-world texture image

this intrinsic component will therefore produce a set of features which are invariant to rotations, reflections and these grey-scale transformations.

The partition of the intrinsic component of jet space which defines the Basic Image Features is based on deciding which type of symmetry of the local image geometry is most nearly consistent with the local jet.

Symmetry of a structure is always relative to some class of transformations. A structure possesses a symmetry when the action of a group of these transformations leaves it indistinguishable from the original; in which case this group is referred to as an automorphism group of the structure. The types of transformations which we consider for images are *image isometries* (Griffin 2008): spatial isometries combined with intensity isometries. We have employed a method to determine all possible automorphism groups (i.e. all potential symmetries) of 2D images relative to this class of image isometries, excluding cases containing discrete periodic translations (which cannot be detected locally) (Griffin et al. 2009).

We say that a filter is *sensitive* to a symmetry if it gives the same response to all images that have that symmetry. As an example, consider a $+1/-1$ filter: if this filter is positioned so that it straddles a putative line of reflection then a necessary criterion for the symmetry is that the filter gives a zero response. Based on this definition, a test has been developed (Griffin and Lillholm 2008) which makes it possible to determine whether a given filter is sensitive to a certain local symmetry.

By combining these elements, we have determined which filters in the span of the second order Gaussian derivative family of Fig. 2 (i.e. which linear combinations of the filters) are sensitive to each of the possible image symmetries; and hence the symmetry sensitivities of the entire second order jet. This allows the regions of the intrinsic component of jet space which represent each type of image symmetry to be identified.

Since most image structures are not perfectly symmetrical, we base our partitioning scheme on deciding which symmetry *most approximately* holds. By selecting an appropriate subset of symmetry types (which deals with the problem of some automorphism groups being subgroups of others) and partitioning the intrinsic component of jet space into Voronoi cells around their corresponding regions using a metric induced by the filter response space (Griffin 2007), we achieve this approximate symmetry classification.

2.1 A BIF-based Texture Representation

By providing a natural quantisation of filter response space into qualitatively distinct types of local image structure, with an appropriate set of in-built invariances, BIFs offer a basis for a viable mathematical alternative to visual dictionaries based on clustering. As discussed above, the advantages of this include avoidance of biases introduced by the clustering algorithm; elimination of a clustering pre-training step; and computational efficiency since image locations are classified into BIFs simply, using Algorithm 1, rather than by a costly nearest-neighbour computation.

However, simply modelling an image as a histogram over our 7 categories produces too coarse a representation. Using a simple 7-bin BIF-histogram texture representation and the classification framework of Sect. 3, only 65% of images from the CUReT dataset are classified correctly; state-of-the-art approaches score in the high nineties percent (see Sects. 3 and 4). We need a way of combining this seven letter 'alphabet' into a sufficiently descriptive collection of 'words' to make up our dictionary.

One way to achieve this is to look at local *configurations* of BIFs, i.e. how the type of local structure in the image *changes* with location and/or scale. The configuration which we evaluate in this paper is a stack of BIFs calculated, at the same spatial location, across four octave-separated scales. We refer to these 'scale templates' as *BIF-columns*, and define σ_{base} to be the finest scale in a BIF-column. Informally, we have found that this selection of four scales seems to produce a representation which captures the right trade-off between specificity and generality. By considering how BIFs vary over scale, rather than space, we retain the rotation-invariance of BIFs, which has been shown (Varma and Zisserman 2005) to be advantageous for texture classification.

The single parameter ε of Algorithm 1 controls how much 'noise' is tolerated before a region is no longer considered sufficiently uniform to be assigned to the 'flat' (pink) BIF category, and is given another label. For texture analysis we do not want any 'flattening' of potentially informative low-contrast structure and so we set $\varepsilon = 0$ (experimental results confirm that increasing ε degrades performance), with the result that this BIF is never selected. Hence we reduce our alphabet to six letters, resulting in a visual dictionary of $6^4 = 1296$ BIF-columns. In practice this also means that we need not compute responses to our zeroth order filter, so assignment of image points to BIF-columns is fully determined by the responses of $5 \times 4 = 20$ filters.

A texture image is thus represented as a 1296-dimensional histogram of BIF-columns. We populate this histogram by counting occurrences of BIF-columns at every pixel in an image, rather than at keypoints. Further, we include description at points which are too close to the edge of the image to accommodate the full spatial support of the filters. Where full support is unavailable, we compensate by wrapping around to the opposite edge of the image. Traditionally, this ought to decrease the accuracy of our models. However, we have observed the opposite: that removing edge-points from our description degrades classification performance to a similar degree to removing the same number of points at locations randomly sampled from across the image. We offer the explanation that this result is a combination of (i) the effects of poorer sampling when these points are removed, with (ii) sufficient homogeneity in the images which we have analysed so that they can reasonably be treated as cyclical.

Thus our texture representation at scale σ_{base} comprises:

1. Compute a stack of four BIF-images at scales σ_{base}, $2\sigma_{\text{base}}, 4\sigma_{\text{base}}, 8\sigma_{\text{base}}$ by convolving the image with a second-order family of Gaussian derivative filters and applying Algorithm 1 (with $\varepsilon = 0$). Transpose to form an array of BIF-columns representing each image pixel.
2. Populate a 1296-bin histogram representation by counting occurrences of BIF-columns over the whole image.

3 Evaluation

We test our BIF-column texture representation by classifying images from the CUReT dataset (Dana et al. 1999). CUReT consists of 61 texture classes each containing 205 images of a physical texture sample photographed under a (calibrated) range of viewing and lighting angles, but without significant variation in scale or in-plane rotation. CUReT is a challenging test of local image description because of the significant intra-class changes in appearance resulting from varying directional light falling on the 3D texture samples. In line with other classification studies using CUReT, we consider only the 92 images per class which afford the extraction of a 200×200 pixel foreground region of texture.

Since our focus is on representation, we use a simple nearest-neighbour classifier rather than a more sophisticated classifier such as support vector machines which has been shown to produce superior results (Caputo et al. 2005; Hayman et al. 2004; Zhang et al. 2006) but requires more tuning of parameters. The classifier is trained by computing representation histograms of all images in the training set; and a novel image classified according to the shortest distance from its representation to each stored training histogram. The most commonly used histogram comparison metric for this purpose is the χ^2 statistic, although others such as a log-likelihood measure have been used (Ojala et al. 2002). We employ a simplified form of the Bhattacharyya distance, $1 - \sqrt{g}.\sqrt{h}$, which is theoretically better suited than χ^2 to calculating distances between distant points in high dimensional space (Thacker et al. 1997). However, we have also experimented with the χ^2 metric in a limited set of experiments and have found no significant difference in the results produced. One possible cause for this is that in a nearest-neighbour classifier all but the smallest distances are effectively ignored and, for small distances, the Bhattacharyya measure approximates the χ^2 measure (Thacker et al. 1997).

For our BIF-column representation, we set the single scale parameter $\sigma_{\text{base}} = 1$ (a multi-scale approach is developed in Sect. 4).

We compare histograms of BIF-columns with four other state-of-the-art histogram representations, using the same classification framework in each case. These are:

VZ-MR8 (610 textons) (Varma and Zisserman 2005): After being grey-scale normalised, images are probed locally using the (normalised) MR8 filter bank, which consists of a Gaussian; a Laplacian of Gaussian; and collections of elongated first order and second order Gaussian derivative filters, each at three scales and six orientations of which only the response with greatest magnitude at each scale is recorded. Thus filter response vectors are eight dimensional in total (although 38 filters are computed in their calculation), are approximately invariant to rotation and, like BIF-columns, describe the local deep structure of an image. To generate a dictionary of textons, filter responses densely sampled from 13 randomly selected images per texture class are clustered using k-means to produce 10 cluster-centre textons per class. Aggregated over the 61 CUReT classes, these 610 textons Voronoi-partition feature space.

VZ-MR8 (2440 textons) (Varma and Zisserman 2005): As VZ-MR8 (610 textons) above, except that 40 cluster-centre textons are learnt per CUReT category resulting in a 2440 dimensional representation.

Table 1 Invariances and extraction costs of features used by the methods compared in Fig. 4

	Invariance to grey-scale transformations	Invariance to spatial transformations	Overview of feature extraction costs at each point
BIF-columns	Affine intensity transformations	Continuous rotations and reflections	Inner product with $5 \times 4 = 20$ Gaussian derivative filters; run Algorithm 1 four times
VZ-MR8 (610 or 2440 textons)	Affine intensity transformations	Small discrete rotations; hence approximately invariant to continuous rotations	Inner product with 38 Gaussian derivative filters; normalise 8 values; calculate 610 (or 2440) distances in 8-d space and select the smallest
VZ-Joint 7×7	Affine intensity transformations	None	Extract values at 49 pixels; normalise 49 values; calculate 610 distances in 49-d space and select the smallest
$\mathrm{LBP}^{riu2}_{24,3}$	Any monotonic grey-scale transformation	Small discrete rotations; hence approximately invariant to continuous rotations	Extract values at 25 interpolated points; 24 subtractions; take sign of 24 values; one query of a lookup table
Equally-spaced histogram bins (11 bins per dimension)	None	None	Inner product with 5 Gaussian derivative filters; up to $5 \times (11 - 1) = 50$ comparisons to select correct bin

VZ-Joint 7×7 (Varma and Zisserman 2009): After being grey-scale normalised, images are described locally by the collected grey-scale values of a 7×7 pixel image patch. The resulting 49-dimensional feature space is partitioned into 610 textons using clustering in the same way as for VZ-MR8 (610 textons).

$\mathrm{LBP}^{riu2}_{24,3}$ (Ojala et al. 2002): Rotation-invariant uniform Local Binary Patterns as described in Sect. 1.4, with 24 points sampled around a circle of radius 3 pixels, resulting in a 26-dimensional representation. Note the low-dimensionality of this representation compared to the others tested.

In addition, we adapt the equally spaced histogram bin representation of Varma and Zisserman (2004) (see Sect. 1.1) to provide a baseline for performance using our filter bank. Ideally, we would divide each of the 20 dimensions (5 filters at each of 4 scales) of our filter response space into n equally spaced bins and represent images as n^{20}-dimensional histograms over their product. However, for any more than 2 bins per dimension this produces representations which are much higher-dimensional than the point at which performance in Varma and Zisserman (2004) begins to deteriorate due to overfitting of data. Thus any representation of this kind would suffer from one or both of overly crude binning and overfitting to noise. Instead, we limit our filters to a single scale ($\sigma = 1$) resulting in a 5-dimensional feature space. Equally spaced binning then produces a n^5-dimensional representation. Experiments show that the best performance is achieved when $n = 11$.

The invariances and extraction costs of each of the six (including BIF-columns) representations which we test are summarised in Table 1. LBPs are cheapest to extract; methods based on clustering are the most expensive.

In each case, as with our BIF-column representation, we count features at every point in an image, and for points which are too close to the edge of the image to accommodate the full spatial support of the filters we compensate by wrapping around to the opposite edge of the image.

Our classification task consists of training with a given number of images randomly chosen from each texture class and assigning all of the remaining images to one of the 61 categories. We repeat this experiment with 100 different random selections of training and test data (as in Zhang et al. 2006) and report the mean fraction of images correctly classified along with the standard deviation. Figure 4 shows results for a range of training set sizes.

First, note that the performance ranking of the six representations tested remains the same regardless of the number of images in the training set. This can be seen as confirming the uncommitted nature of the nearest neighbour classifier used with each of the representations. BIF-columns score highest, followed by the two MR8-based representations (with the richer 2440-bin representation slightly superior) and then 7×7 image patches. The performance of uniform Local Binary Patterns is significantly below those of the other approaches (apart from the baseline equally-spaced histogram bin representation) for all but the smallest collections of training images. However, it should be noted that this representation is only 26-dimensional, compared to a minimum of 610 dimensions for other methods. This reflects its design goal of being able to cope with smaller

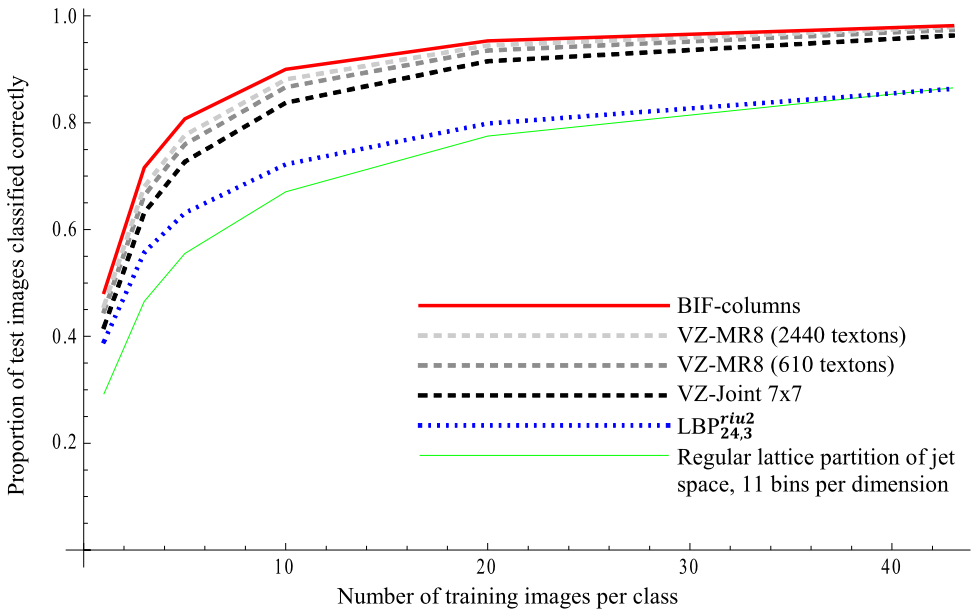


Fig. 4 The mean proportion of correctly classified images over 100 random splits of the CUReT dataset into training/test data, for a range of training set sizes. The best result for BIF-columns (with 43 training images per class) is 98.1±0.3%

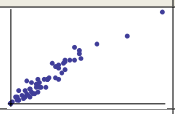
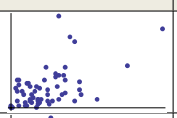
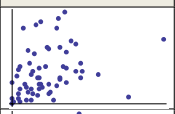
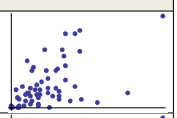
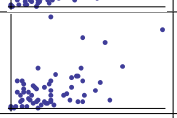
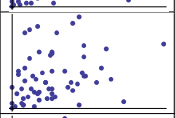
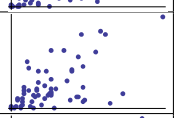
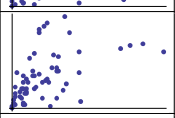
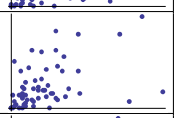
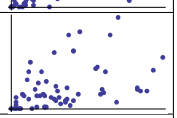
	VZ-MR8 (610 textons)	VZ-MR8 (2440 textons)	VZ-Joint 7x7	$LBP^{riu2}_{24,3}$	BIF-columns
VZ-MR8 (610 textons)					
VZ-MR8 (2440 textons)	0.96				
VZ-Joint 7x7	0.55	0.54			
$LBP^{riu2}_{24,3}$	0.30	0.29	0.52		
BIF-columns	0.56	0.59	0.46	0.41	

Fig. 5 Correlation between the marginal distributions by class of incorrectly classified images (taken across all 100 training/test splits with 43 training images per class) for each pair of representations. There is strong correlation between the two representations using the same (MR8) local description, but only weak correlation between other representations. The strongest correlation for LBPs is with VZ-Joint (although the converse is false). This could be explained by the relative similarity of these two representations in using greyscale-based descriptions and in the extents of their local regions of support, despite the very different forms of their feature space quantisation. Similarly, the strongest correlation for BIF-columns is with the two MR8-based methods, which also probe the image using Gaussian derivative filters

images: fewer bins produce a less precise representation but one which can be populated more accurately when the quantity of data available is a limiting factor. However, the proximity of the two MR8-based approaches suggest that the dimensionality of representation is not a major cause of variation in performance between the other four (more consonant) representations.

The relative similarity in performance of the best four methods for large numbers of training images begs the question of whether we are pushing against a ceiling of a minority of images which are particularly difficult for histogram-based texture representations to cope with. Figure 5 suggests that this is not the case: although there is some correlation between the distributions of images misclassified by different representations, in the majority of cases it is fairly weak, i.e. in general different representations mis-classify different images. One notable exception to this is the strong correlation between the two representations using the same (MR8)

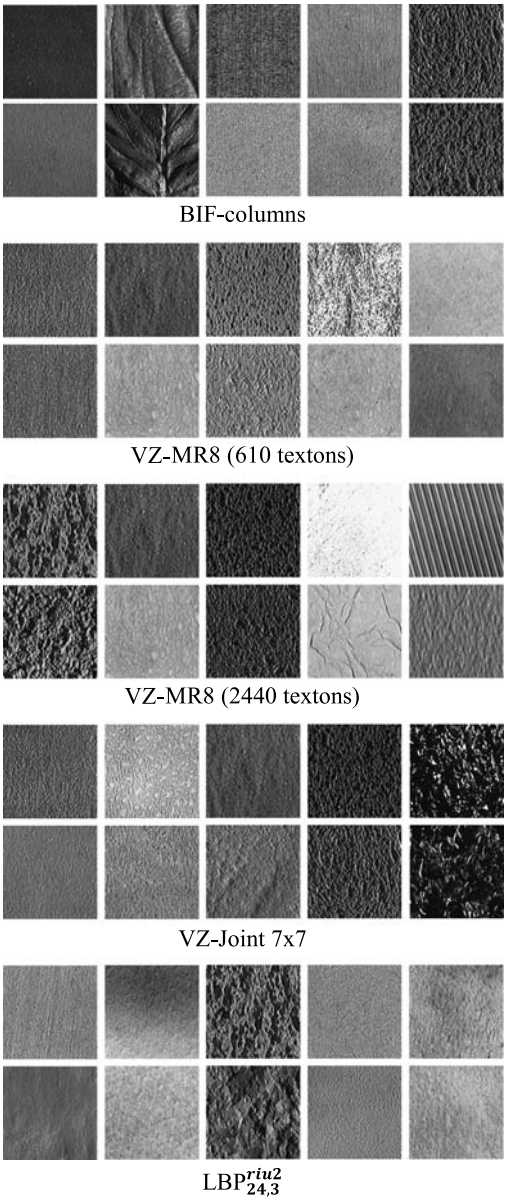


Fig. 6 The (1st, 3rd, 5th, 7th and 9th) most frequently misclassified images over the 100 trials (*top*), and the images for which they were most often mistaken (*bottom*). For each representation, some images are perceptually similar to those for which they are mistaken and some are not

local description, which differ only in the number of cells into which their feature spaces are partitioned. The particular types of texture which appear problematic for each representation defy easy characterisation (Fig. 6).

4 Multi-scale Histogram Matching

Although our representation describes the local deep structure in an image, it is not scale invariant. The scale of the base of our BIF-columns, σ_{base}, remains fixed. In order to be able to usefully describe sets of textures which, unlike CUReT, contain significant variation in scale, we extend our representation and introduce a multi-scale histogram comparison.

There are two related problems which should be addressed in an appropriate scale-treatment of texture. First, images of the same texture should be recognised as such despite being taken from different distances (scale invariance). Second, the texture representation should incorporate description at (and representations should be compared across) a range of scales (referred to elsewhere Ojala et al. 2002 as multi-resolution analysis); rather than at one fixed scale which is chosen as a compromise for the given dataset, or at one intrinsic scale. This ensures (i) that the image is probed at scales matching those of important local structure in that image, and (ii) that where (as frequently happens) images contain informative structure at a number of scales, full use is made of this information: rather than, for example, having to choose whether a brick wall is best represented by the layout of the bricks or the microstructure of the clay.

Hayman et al. (2004) adopt a pure learning approach which addresses the first of these problems (and, to an extent, part (i) of the second) by, in effect, augmenting the training data with artificially rescaled versions of the original training images. By decoupling the descriptions of textures at each scale it makes the implicit assumption that textures need only be matched at a single dominant intrinsic scale; thus although it works well for the datasets tested, it may not extend to the more general problem.

Our approach retains the links between representations of the same texture at different scales by modelling an image as a stack of BIF-column histograms computed over a range of scales (indexed by σ_{base}) in the same way as for the single-scale representation described in Sect. 2.1. The range of σ_{base}s which we have found to be effective (as a trade-off between descriptiveness and computational complexity) increment in quarter-octaves from $2^{-1/4}$ to $2^{6/4}$ meaning that, with the four-octave span of our BIF-columns, the total range of scales analysed runs from 0.84 to 22.6 pixels. We emphasise the difference between BIF-columns which describe the *local* variation in structure around some point in scale space; and histogram stacks which represent the *global* variation over scale of the texture itself.

The second of the above criteria is then addressed by comparing histogram-stack texture representations using a *multi-scale metric*, based on the Bhattacharyya distance, which computes a weighted average of the distances between histograms at each scale. The first criterion (scale invariance) is realised by allowing histogram stacks to be shifted in scale relative to one another before calculation of this distance, as shown in Fig. 7 (*scale-shifting*).

More specifically, to compare stacks of normalised BIF-column histograms for images A and B, calculated at

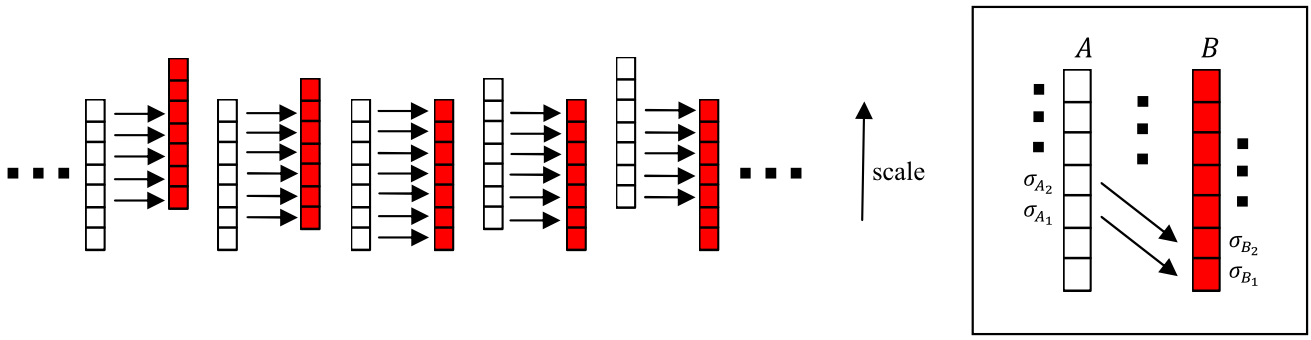


Fig. 7 Multi-scale comparison of images A and B. *Left*: Scale shifting: Histogram stacks containing n histograms are shifted relative to each other in scale in each of $2n-1$ possible ways, to allow matching of similar features appearing at different scales in each image. *Right*: The notation used in (1)

column-base scales $\sigma_{\text{base}} = \sigma_{A_1}, \sigma_{A_2}, \ldots, \sigma_{A_n}$ and $\sigma_{B_1}, \sigma_{B_2}, \ldots, \sigma_{B_n}$ respectively (see Fig. 7, right), our multi-scale metric calculates a weighted average of squared Bhattacharyya distances computed at each pair of base scales $(\sigma_{A_i}, \sigma_{B_i})$,

$$\frac{\sum_{i=1}^{n} \frac{(1-\sqrt{h(A;\sigma_{A_i})}.\sqrt{h(B;\sigma_{B_i})})^2}{\sigma_i^2}}{\sum_{i=1}^{n} \frac{1}{\sigma_i^2}} \tag{1}$$

where $h(I;\sigma_j)$ is the normalised BIF-column histogram of image I computed at base scale σ_j and $\sigma_i^2 = \sigma_{A_i}^2 + \sigma_{B_i}^2$. The weighting by $\frac{1}{\sigma_{A_i}^2+\sigma_{B_i}^2}$ discriminates against poorly sampled coarse scale representations. Normalisation commensurates distances for differently shifted comparisons, allowing the multi-scale scheme to be incorporated directly into our nearest neighbour classifier: the distance between two images is effectively taken to be the minimum of the distances calculated between those images in each of the $2n-1$ possible ways illustrated in Fig. 7.

This multi-scale classifier is computationally expensive compared to our single scale classifier. Instead of using a single Bhattacharyya distance to compare two images, for histogram stacks containing n histograms it is necessary to calculate $2\sum_{i=1}^{n-1} i + n = n^2$ distances. Hence with our stacks of 8 histograms, we calculate 64 times as many distances as for our single scale classifier. However, in practice calculation of these distances is fast and the additional time taken is small compared to the time to calculate a representation from an image.

4.1 Evaluation

We have tested our multi-scale scheme by classifying texture images from three datasets: the CUReT dataset as used in Sect. 3, KTH-TIPS (Hayman et al. 2004) and UIUCTex (Lazebnik et al. 2003). We emphasise that our method is exactly the same for each dataset, with no tuning of parameters.

The KTH-TIPS dataset extends CUReT by imaging new samples of 10 of the CUReT textures at a subset of the viewing and lighting angles used in CUReT but also over a range of scales, producing 81 200 × 200 pixel images per class. Although KTH-TIPS is designed in such a way that it is possible to combine it with CUReT in testing, we follow Zhang et al. (2006) in treating it as a stand-alone dataset. UIUCTex contains 25 classes, each of 40 640 × 480 pixel images. The dataset is uncalibrated and classes contain images taken at a variety of scales and viewpoints, and sometimes with non-rigid deformations of the samples. However, variations in lighting geometry are less severe than for the other two datasets.

As in Sect. 3, results are reported as the mean proportion of images correctly classified over 100 random splits into training and test data, along with one standard deviation. We use 43, 40 and 20 training images per class respectively for CUReT, KTH-TIPS and UIUCTex.

Results (as reported in Crosier and Griffin 2008) are shown in Table 2. Despite not being modified to suit each dataset, our multi-scale BIF-columns scheme scores well across all three datasets, producing what we believe to be the best reported result on the KTH-TIPS images; equal to the best reported results on UIUCTex; and the best reported results out of those which use a nearest-neighbour classifier on CUReT. The overall best performance on CUReT is from Broadhurst's conference paper (Broadhurst 2005), which achieved 99.22% correct classification using a Gaussian Bayes classifier with marginal filter distributions.

Analysis of the behaviour of our multi-scale approach shows that the two component parts—the multi-scale comparison metric and histogram-stack scale-shifting—complement each other appropriately (Fig. 8). Our multi-scale metric improves performance over our single scale scheme on both the UIUCTex and CUReT datasets, confirming that texture comparison at a range of scales is important even in the absence of significant intra-class variation in scale; where as the scale-shifting part of our algorithm is useful only

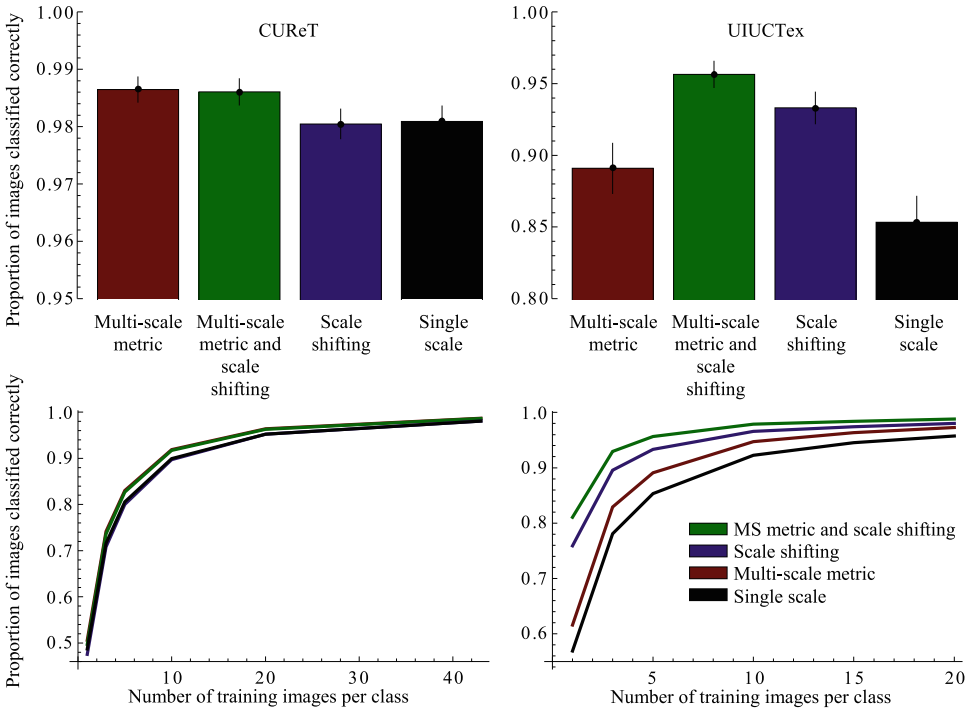


Fig. 8 The proportion of images correctly categorised by each of the two components of our multi-scale classifier (the multi-scale metric and scale-shifting); our full multi-scale classifier (these components combined); and our single scale classifier as evaluated in Sect. 3. *Top*: Using 43 training images per CUReT class and 5 training images per UIUCTex class. *Bottom*: The trend as the number of training images per class varies. In each case, we report the mean and standard deviation over 100 trials of the fraction of remaining images correctly classified. For CUReT, which does not contain significant intra-class variation in scale, there is no benefit to be gained by using scale shifting. However, comparing images at a range of scales using our multi-scale metric does result in improved performance, suggesting that images contain informative structure at multiple scales. For UIUCTex, which does contain significant intra-class scale variations, both the multi-scale metric and scale-shifting produce improvements over our single scale classifier, with the combination of the two in our full multi-scale scheme giving the best performance. For both datasets, these results hold independently of the number of training images per class

Table 2 Classification scores on the CUReT, UIUCTex and KTH-TIPS datasets. Scores are as originally reported, except for those marked † which are taken from the comparative study in Zhang et al. (2006)

	CUReT 43 training images per class	UIUCTex 20 training images per class	KTH-TIPS 40 training images per class
Multi-scale BIF-columns	$98.6 \pm 0.2\%$	$\mathbf{98.8 \pm 0.5\%}$	$\mathbf{98.5 \pm 0.7\%}$
Varma & Zisserman—MR8 (Varma and Zisserman 2005)	97.43%		
Varma & Zisserman—Joint (Varma and Zisserman 2009)	98.03%	$97.83 \pm 0.66\%$	$92.4 \pm 2.1\%$†
Hayman et al. (2004)	$98.46 \pm 0.09\%$	$92.0 \pm 1.3\%$†	$94.8 \pm 1.2\%$†
Lazebnik et al. (2003)	$72.5 \pm 0.7\%$†	96.03%	$91.3 \pm 1.4\%$†
Zhang et al. (2006)	$95.3 \pm 0.4\%$	$\mathbf{98.70 \pm 0.4\%}$	$95.5 \pm 1.3\%$
Broadhurst (2005)	$\mathbf{99.22 \pm 0.34\%}$		
Varma and Ray (2007)		$\mathbf{98.9 \pm 0.68\%}$	

when scale-invariance is called for. Indeed, for the CUReT data, distances between matching images are nearly always smaller when no scale-shifting is used, meaning that, for these images, our multi-scale algorithm rarely acts any differently than if this component were absent (Fig. 9). By contrast, shifting occurs frequently for UIUCTex images. That is, most of the time, scale-shifting is *used* only when scale-invariance is called for.

Note in Fig. 8 that, on the UIUCTex images, the method which produces the next-best results to our full multi-scale

scheme is scale-shifting without the multi-scale comparison metric, which is the method most similar to Hayman et al.'s approach (Hayman et al. 2004).

Figure 10 shows examples of images which are mis-classified by our multi-scale scheme.

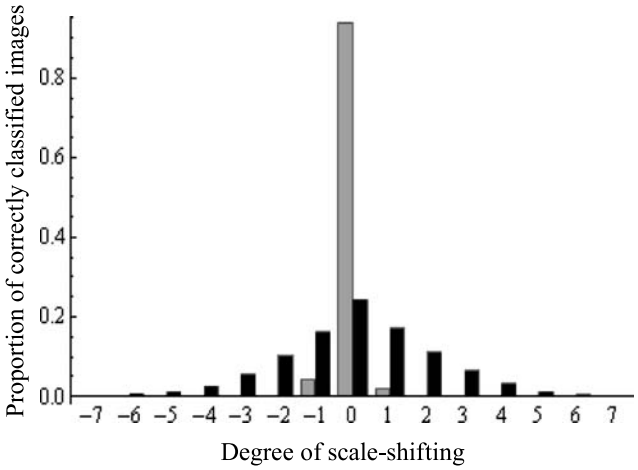


Fig. 9 The proportion of images, out of those which are correctly classified, in which the nearest training image representation is found using the given degree of histogram-stack scale-shifting (Fig. 7), for CUReT (*grey*) and UIUCTex (*black*) images. For CUReT, the distance calculated between histogram-stack representations after shifting is nearly always larger than the distance calculated with no shifting, i.e. the closest training image is most frequently found using no scale-shifting: as is appropriate in the absence of intra-class scale changes. For UIUCTex, which does contain intra-class variations in scale, it is more common for a distance calculated after shifting to be smaller than the distance with no shifting, resulting in a flatter distribution

4.2 Generality

In this section we investigate whether this multi-scale histogram matching approach is applicable to other histogram texture representations.

In particular, we implement a multi-scale extension to the VZ-MR8 representation (Varma and Zisserman 2005) described in Sect. 3. To recap, images are probed using the MR8 filter bank which consists of a Gaussian; a Laplacian of Gaussian; and collections of elongated first order and second order Gaussian derivative filters, each at three scales and six orientations of which only the response with greatest magnitude at each scale is recorded. The filters range in scale from 1 to 12 pixels. Filter responses sampled from a subset of training images for each class are clustered using k-means to produce 10 cluster-centres per class, and the aggregate of these cluster centres over all classes is used to Voronoi-partition the 8-dimensional feature space into textons. An image is then represented as a histogram by counting the textons corresponding to the filter responses calculated at each pixel. As for our single-scale implementation, for points which are too close to the edge of the image to accommodate the full spatial support of the filters we compensate by wrapping around to the opposite edge of the image.

As with BIF-columns, the scale of representation can be indexed by a factor σ_{base} by which the scales of all filters are multiplied (or, equivalently, by which the image is Gaussian-blurred prior to filtering). The same range of σ_{base}s are used as for BIF-columns. This leaves the question of how to choose a multi-scale set of textons so that representations at each scale are over the same visual dictionary, in

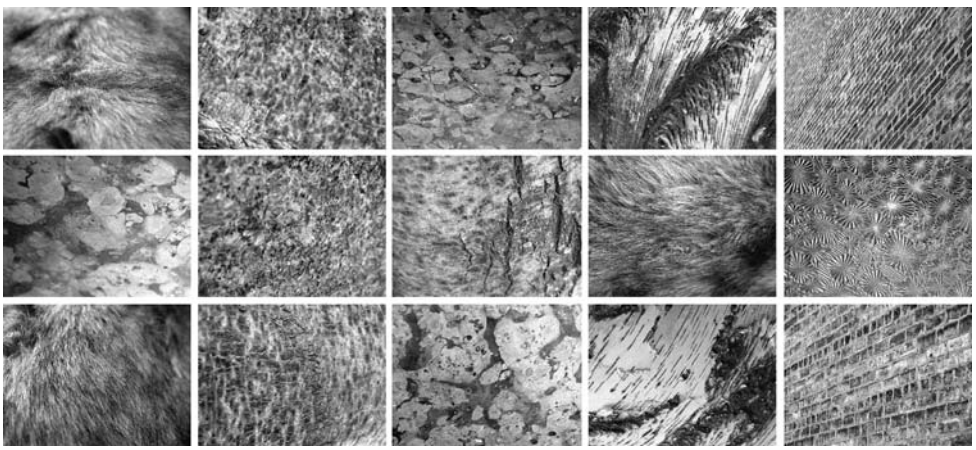


Fig. 10 Examples of images from the UIUCTex dataset which are mis-classified by our multi-scale algorithm (*top*); the training images for which they are most often mistaken (*centre*); and the most frequently corresponding 'nearest misses' from the correct class (*bottom*). *Left to right*, the images are the first ('fur' mistaken for 'marble'), second ('bark 2' mistaken for 'granite'), third ('marble' mistaken for 'bark 2'), fourteenth ('bark 3' mistaken for 'fur') and seventeenth ('brick 1' mistaken for 'glass 1') most frequently mis-classified UIUCTex images, counted over 100 random splits into 20 training and 20 test images per class. Mis-classified images are often perceptually similar, on a local level, to those for which they are mistaken, as in the middle three examples. However, the most frequently mis-classified image (*left*) bears little resemblance to the training image selected by our algorithm. The *right-most* example demonstrates a lack of sensitivity to the regularity property of the brick texture, a limitation inherent in the representation of images as histograms

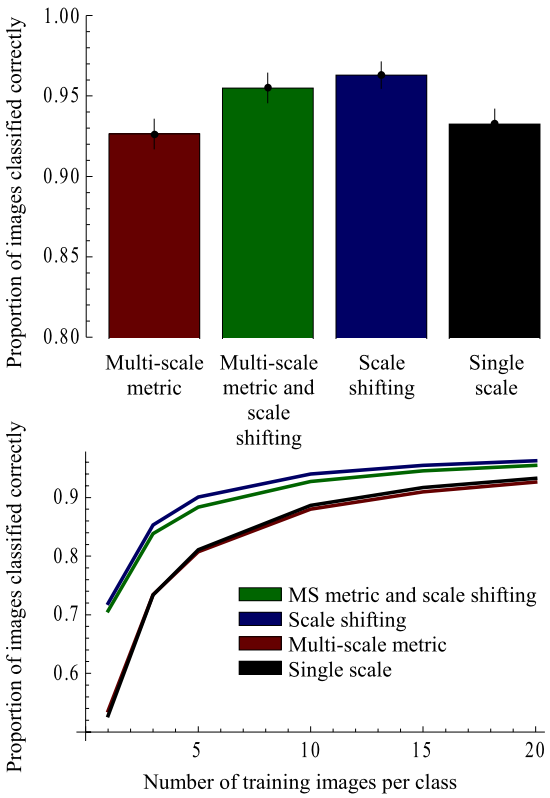


Fig. 11 For our multi-scale extension to the VZ-MR8 texton-histogram representation (Varma and Zisserman 2005), the proportion of UIUCTex images correctly categorised by each of the components of our multi-scale classifier. *Top*: Using 20 images per class. *Bottom*: The trend as the number of training images per class varies. Scale-shifting produces improvements over our single-scale classifier, but our multi-scale metric consistently degrades performance in this situation

order to allow cross-scale comparison. Simply learning histograms at a single σ_{base} results in poor representations at other σ_{base}s. A natural choice is to learn textons by clustering feature responses at a range of scales together: for each class, we randomly select 13 training images and aggregate filter responses for all of these images at every scale. This results in a very large number of datapoints to be clustered using k-means, so these are randomly subsampled to produce the same number of filter responses as for a single scale (i.e. the number of pixels within the 13 images). Having represented images as scale-stacks of histograms over this set of textons, our scale-shifting and multi-scale metric can be used without further alterations.

We have tested this multi-scale texton approach by classifying images from the UIUCTex dataset. The best classification score achieved is $96.3 \pm 0.8\%$ using scale-shifting alone (Fig. 11). Although scale-shifting improves performance, for this representation our multi-scale metric is a handicap. The mechanism for this drop in performance is unclear. However, it does show that, whilst our multi-scale classifier performs well with our BIF-column representation, it is somewhat dependent on the choice of representation and is not therefore appropriate in every case.

5 Summary

We have developed a statistical texture representation which models images as histograms over a dictionary of features which is based on the qualitative type of local geometric structure, encoded by Basic Image Features, rather than a dictionary based on clustering. Our features are naturally invariant to rotation and reflection, and addition and linear multiplication of illumination intensity; and we have extended the approach to incorporate invariance to changes in scale.

Our approach has the advantages over methods which use clustering of simplicity—there is no need for a pre-training step to learn a visual dictionary—and computational efficiency, since we assign feature vectors to histogram bins without needing to perform a nearest-neighbour computation. In addition, it avoids the potential introduction of biases by clustering algorithms poorly suited to the data.

We have tested our implementation on three popular and challenging texture datasets and find that it produces consistently good classification results on each, including what we believe to be the best reported for the KTH-TIPS and equal best reported for the UIUCTex databases. Further, it does this without requiring modification or tuning of parameters between datasets.

Acknowledgement EPSRC-funded project 'Basic Image Features' EP/D030978/1.

References

Broadhurst, R. E. (2005). Statistical estimation of histogram variation for texture classification. In *Proceedings of the fourth international workshop on texture analysis and synthesis* (pp. 25–30). Beijing, China, October 2005.

Caputo, B., Hayman, E., & Mallikarjuna, P. (2005). Class-specific material categorisation. In *Tenth IEEE international conference on computer vision, 2005. ICCV 2005* (Vol. 2, pp. 1597–1604).

Crosier, M., & Griffin, L. D. (2008). Texture classification with a dictionary of basic image features. In *IEEE conference on computer vision and pattern recognition 2008* (pp. 1–7), June 2008.

Cula, O. G., & Dana, K. J. (2001). Recognition methods for 3d textured surfaces. In *Proceedings of SPIE conference on human vision and electronic imaging VI*, San Jose, 2001.

Dana, K. J., Van-Ginneken, B., Nayar, S. K., & Koenderink, J. J. (1999). Reflectance and texture of real world surfaces. *ACM Transactions on Graphics*, *18*(1), 1–34.

Freeman, W. T. & Adelson, E. H. (1991). The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13*(9), 891–906.

Griffin, L. D. (2007). The 2nd order local-image-structure solid. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(8), 1355–1366.

Griffin, L. D. (2008). Symmetries of 1-d images. *Journal of Mathematical Imaging & Vision*, *31*(2–3), 157–164.

Griffin, L. D. & Lillholm, M. (2007). Feature category systems for 2nd order local image structure induced by natural image statistics and otherwise. *Proceedings—SPIE*, *6492*(09), 1–11.

Griffin, L. D., & Lillholm, M. (2008). Classifying local image symmetry using a colocalised family of linear filters. Perception 37 ECVP Abstract Supplement 2008.

Griffin, L. D., Lillholm, M., Crosier, M., & van Sande, J. (2009). Basic image features (BIFs) arising from local symmetry type. In *LNCS: Vol. 5567. Proceedings SSVM '09* (pp. 343–355). Berlin: Springer.

Harris, C. G., & Stephens, M. (1988). A combined corner and edge detector. In *Fourth Alvey vision conference* (pp. 147–151), Manchester, UK.

Hayman, E., Caputo, B., Fritz, M., & Eklundh, J. (2004). On the significance of real-world conditions for material classification. In *ECCV 2004* (pp. 253–266).

Jurie, F., & Triggs, B. (2005). Creating efficient codebooks for visual recognition. In *Tenth IEEE international conference on computer vision, 2005. ICCV 2005* (Vol. 1, pp. 604–610).

Konishi, S., & Yuille, A. L. (2000). Statistical cues for domain specific image segmentation with performance analysis. In *Proceedings of IEEE conference on computer vision and pattern recognition, 2000* (Vol. 1, pp. 125–132).

Lazebnik, S., Schmid, C., & Ponce, J. (2003). A sparse texture representation using affine-invariant regions. In *Proceedings of 2003 IEEE computer society conference on computer vision and pattern recognition, 2003* (Vol. 2, pp. II-319–II-324).

Leung, T., & Malik, J. (2001). Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, *43*(1), 29–44.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *The proceedings of the seventh IEEE international conference on computer vision, 1999* (Vol. 2, pp. 1150–1157).

Ojala, T., Maenpaa, T. & Pietikainen, M. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Transactions on Pattern Analysis and Machine Intelligence*, *24*(7), 971–987.

Thacker, N. A., Aherne, F. J., & Rockett, P. I., (1997). The Bhattacharyya metric as an absolute similarity measure for frequency coded data. *Kybernetika*, *34*(4), 363–368.

Varma, M., & Garg, R. (2007). Locally invariant fractal features for statistical texture classification. In *IEEE 11th international conference on computer vision, 2007. ICCV 2007*.

Varma, M., & Ray, D. (2007). Learning the discriminative power-invariance trade-off. In *IEEE 11th international conference on computer vision, 2007*.

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

Varma, M. & Zisserman, A. (2004). Unifying statistical texture classification frameworks. *Image and Vision Computing*, *22*(14), 1175–1183.

Varma, M. & Zisserman, A. (2005). A statistical approach to texture classification from single images. *International Journal of Computer Vision*, *62*(1), 61–81.

Zhang, J., Marszalek, M., Lazebnik, S., & Schmid, C. (2006). Local features and kernels for classification of texture and object categories: a comprehensive study. In *Conference on computer vision and pattern recognition workshop, 2006* (p. 13).