Texture Classification with a Dictionary of Basic Image Features

Michael Crosier
University College London
London, UK
m.crosier@cs.ucl.ac.uk

Lewis D Griffin
University College London
London, UK
l.griffin@cs.ucl.ac.uk

Abstract

Many successful recent approaches to texture classification model texture images as distributions over a set of discrete features, or textons, which correspond to a partitioning of the space of responses to local descriptors such as filter banks or image patches. This partitioning is learned by unsupervised clustering of descriptor responses taken from the dataset to be analysed. Here, we explore a quantization of filter responses into a dictionary of discrete features which is based on geometrical, rather than statistical, considerations, resulting in a simple texture description based on a dictionary of 'visual words' which is independent of the images to be described. A multi-scale classification scheme built on this dictionary is evaluated. The results presented are, to the best of our knowledge, state-of-the-art for the UIUCTex and KTH-TIPS datasets, and close to the state-of-the-art for CUReT, despite using a less sophisticated classifier.

1. Introduction

The classification of textures from single images taken under arbitrary (and unknown) viewing and illumination conditions has gained a great deal of attention over recent years.

One approach which has proved particularly successful for this task involves representing images statistically as distributions of local features [1-10]. Such methods can be coarsely characterised by i) how they represent an image and ii) the classifier used. In this paper we are interested in representation. Statistical approaches to texture representation can be grouped by their treatment of a set of sub-problems:

i) Which points in an image should contribute to its representation? [6, 8] select a sparse set of points using Harris-affine or Laplacian detectors, whilst [3, 4, 7, 10] describe every point in an image.

ii) How should these points be described? Much work has focussed on this problem. Amongst others, [1, 4, 5, 7] employ the joint responses of various filter banks, with the most successful being the rotation-invariant MR8 of [4], also used in [7]; [6, 8] use modified SIFT[11] and intensity domain SPIN images; [3] uses a simple greyscale image patch for its Joint descriptor, as well as an MRF representation; and [10] describes local fractal dimension and length.

iii) Which of two paradigms for image representation should be adopted? The choice here is between modelling images as histograms over a dictionary of features, or *textons* [3, 4, 7, 10] (which requires a pre-training step to learn a feature dictionary); or modelling images as signatures [6, 8] of features derived adaptively for each image.

iv) How should responses to descriptors be grouped in order to form a feature dictionary or image signature? The shared tactic of methods which model joint descriptor responses is to use *k*-means clustering to partition descriptor-response space (*feature space*). Hence each feature can be represented by a cluster-centre. For methods within the texton dictionary paradigm, this allows new descriptor values to be assigned to a texton by finding the nearest cluster-centre in feature space. Within the signatures paradigm, being able to compute the ground distance between cluster-centres allows use of the Earth Mover's Distance to measure the similarity between images.

For the texton dictionary case, the pre-training step typically calculates k cluster-centres for each of the C classes to be analysed and then combines them to produce kC textons. However, this may result in an unnecessarily redundant set of features; [4] shows that textons can be combined to produce a smaller set with only a small loss in performance.

Given an image representation, classification commonly proceeds by either nearest-neighbour matching (returning the class of the training image whose representation lies closest – with some metric, usually the χ^2 statistic [3, 4, 10] or Earth Mover's Distance [8] – to that of the test image) or non-linear (kernel-based) SVMs [6, 7]. *k*-nearest neighbours matching has also been explored [7], but produces no improvement over nearest-neighbour. [7] and [6] demonstrate the superiority of

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

SVMs over NN for texture classification. However, nearest neighbour classification remains useful for studies focussing on description and representation due to its relative poverty of implementation choices.

In all of these methods, the partitioning of feature space is accomplished by unsupervised clustering of local descriptor responses using some variation of the *k*-means algorithm. Labels are defined by cluster centres and points are assigned to some label depending on which cluster centre is closest to that point's local descriptor; thus, the clustering induces a Voronoi partition on feature space.

In this paper we explore an alternative way to achieve this partitioning, based on assigning a common label to points which share the same predominant geometrically-defined 'type' of local structure. Using a dictionary of visual words which is pre-defined, rather than being generated from the dataset to be analysed, produces a simpler approach to classification and one which may pick up on useful subtleties which are smoothed over by the clustering process. It also obviates the need to ensure that high dimensional feature spaces are sufficiently well populated to allow accurate clustering.

By 'type' of local structure, we suggest that any local geometric structure can be assigned to one of a finite vocabulary of primitives which are somehow qualitatively distinct. This is the premise of the Geometric Texton Theory of [12], which categorizes local structure based on the local symmetries which it possesses (see section 2). This induces a partition on the filter-response space of six Gaussian derivative filters. We refer to these features as 'Basic Image Features' or BIFs.

1.1. Related work

Aside from the methods detailed above which employ *k*-means clustering to quantize descriptor responses into discrete features over which the texture can be represented statistically, Konishi & Yuille [13] have divided filter-response space into a lattice by adaptively quantizing the value of each of N filters, into b_n bins, separately; before modelling a texture as a distribution over these $\prod_{n=1}^{N} b_n$ bins. As with methods using *k*-means, the partition is determined by analysis of the dataset in question.

Local Binary Patterns [14] resemble our approach in that each LBP scheme constitutes a pre-defined dictionary of features over which images are represented statistically. The main difference lies in the mathematical basis for the choice of feature set, which in our case is geometrical and in the case of uniform LBPs is based on the statistical properties of texture images in general.

In the remainder of this paper, we describe a texture classification system in which images are represented by histograms over a dictionary of combinations of Basic Image Features, rather than cluster-centre textons; present a multi-scale histogram comparison technique; and evaluate categorization performance on a range of popular texture datasets. The results presented are, to the best of our knowledge, state-of-the-art for the UIUCTex and KTH-TIPS datasets, and close to the state-of-the-art for CUReT, despite using a less sophisticated classifier.

2. Basic Image Features

Basic Image Features [12, 15] are defined by a partition of the filter-response space (*jet space*) of a set of six Gaussian derivative filters (Figure 1). This set of filters describes an image locally up to second order at some scale.

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

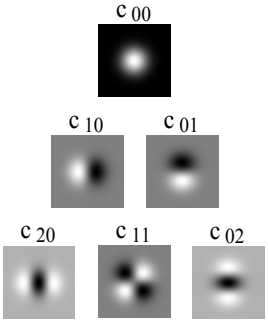

Figure 1: The filter bank used to calculate BIFs, consisting of one zeroth-order, two first order and three second order Gaussian derivative filters, all at the same scale. We refer to the vector of responses as a *local jet*. It describes up to the second order term of the local Taylor series. Note that in our implementation, since we set the parameter ε of Algorithm 1 to be zero, we do not need to compute responses to the zeroth order filter.

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 [16] 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 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 [15].

A test is developed [15] which shows whether a filter is sensitive to a certain local symmetry, i.e. whether it is able to detect invariance under a group of transformations (i.e. a prospective automorphism group). The type of transformations considered are *image isometries* [17]: spatial isometries combined with intensity isometries. The possible automorphism groups of 2D images relative to the class of image isometries, excluding cases containing discrete periodic translations, are determined [15]. Hence we can use our test to decide which filters in the span of the second order Gaussian derivative family of figure 1 (i.e. which linear combinations of the filters) are sensitive to each of these symmetries. (For example, the only such filters which are sensitive to reflectional symmetry in vertical lines (at some scale) are of the form $\alpha\, c_{10} + \beta\, c_{11}$ centred on the line of reflection.) 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 into Voronoi cells around their corresponding regions using a metric induced by the filter response space [16], we achieve this approximate symmetry classification.

i. Measure filter responses c_{ij}

ii. Compute $\lambda = \sigma^2(c_{20} + c_{02})$,
$\gamma = \sigma^2\sqrt{(c_{20} - c_{02})^2 + 4c_{11}^2}$

iii. Classify according to the largest of:
$\left\{\varepsilon c_{00}, 2\sigma\sqrt{c_{10}^2 + c_{01}^2}, \pm\lambda, 2^{-\frac{1}{2}}(\gamma \pm \lambda), \gamma\right\}$

Algorithm 1: Calculation of BIFs. For texture analysis we set the parameter $\varepsilon \to 0$. See figure 2 for an illustration of the types of symmetry captured by each of the seven BIFs.

The algorithm for computing Basic Image Features from filter responses turns out to be very simple, and is given in Algorithm 1. Figure 2 illustrates the types of structure / symmetry which are represented by each of the seven BIFs. One of these (the pink label of Figure 2) represents a degenerate case of 'flat' structure which, given its uniformity, contains every possible kind of local symmetry. For this single category, a tunable parameter ε determines how much 'noise' is tolerated before a region is no longer considered to be flat; and hence is assigned to another category. For texture analysis we do not want any flattening of potentially important low-contrast structure, and so we set this parameter so that this seventh label is never used. Experiments confirm that this also produces the best results. Hereafter, we will refer to a six-feature system of BIFs.

An example of a texture image densely labelled with the six BIF features computed at two different scales is shown in figure 3.

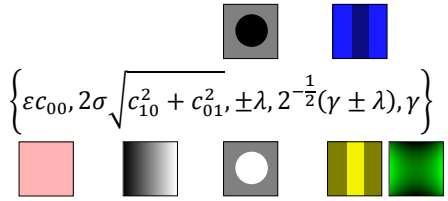

Figure 2: Stereotypical image patches corresponding to each of the seven BIFs defined by step iii of Algorithm 1. Colours correspond to the labelling system of Figure 3.

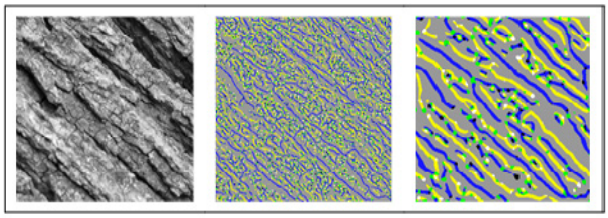

Figure 3: An image of bark from the UIUCTex database [8], with BIFs computed at scales $\sigma = 1$ and $\sigma = 4$. See Figure 2 for a key to the colours used for labelling.

3. Method & Results

3.1. Features

Simply modelling an image as a histogram over six categories would not be expected to be sufficiently descriptive to represent a texture well (and indeed, using histograms of BIFs at a single scale to represent each image produces only 65% correct classification on the CUReT database). In order to produce a less coarse representation we need a way of combining this six letter 'alphabet' into a sufficiently descriptive collection of 'words'. One way to achieve this is to describe local *configurations* of BIFs, i.e. how local structure in the image *changes*.

Varma & Zisserman [2] demonstrated the advantages of rotationally invariant local description. A natural choice is therefore to exploit the rotation-invariance of the BIFs themselves by considering stacks, or 'columns', of BIFs over scale; rather than a non-rotationally invariant spatial configuration of BIFs. So our features describe the change in local structure over scale.

Empirically we have found that a stack of four BIFs distributed logarithmically in scale space over four octaves (i.e. at scales 1,2,4 and 8 times the 'base'), resulting in a 6^4=1296 dimensional image representation, seems to capture the right trade-off between specificity

and generality. For the rest of the paper, these features will be referred to as BIF-columns. However, we do not claim that this is an optimal way of using BIFs for local description.

BIF-columns are illustrated in figure 4. We represent images as distributions over these features, calculated at every pixel.

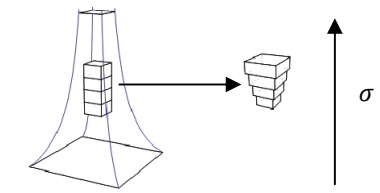

Figure 4. Since sampling in scale space (left) decreases as scale increases, our BIF-columns are actually more like pyramids (right) in that they describe larger local regions of the image at coarser scales.

We compute Gaussian derivative filter responses for an image by convolution in the spatial domain (using the Mathematica implementation from [18]). For points near to the edge of the image, the convolution wraps around to the opposite edge. Ideally, points where the full spatial support of the filter does not lie within the image should be discarded; however, we have found that this consistently degrades performance, regardless of our other implementation choices. We hypothesize that this result is due, on the one hand, to the effects of poorer sampling when these points are removed (and indeed, a similar degradation is produced when we disregard points randomly subsampled from across the image to simulate this reduction in sampling), and on the other hand, the possibility that the images which we have analysed are sufficiently homogeneous that treating them cyclically is not unreasonable. We use filter responses calculated at every point in the image for the results presented here, but note that this deserves further investigation.

Having represented a texture image as a histogram over our dictionary of BIF-columns, we use nearest-neighbour classification (as in [4]) to classify test images. As a histogram comparison metric we employ the Bhattacharyya distance, $1 - \sqrt{g}.\sqrt{h}$, rather than the more commonly used chi-square statistic, since it has been shown to possess superior theoretical properties in certain situations [19]. However, in practice we have found the results produced by both to be very similar.

Thus the single scale version of our method comprises, at scale σ:

Representation:

- Compute a stack of four BIF-images at scales $\sigma, 2\sigma, 4\sigma, 8\sigma$, using Algorithm 1. Transpose to get an array of BIF-columns representing each pixel in the image.
- Count occurrences of each of the 6^4=1296 possible BIF-columns and form a normalised histogram.

Classification:

- Train by computing a BIF-column histogram (above) for each training image.
- Classify by computing a BIF-column histogram (above) for the test image and finding the nearest neighbour of the stored training histograms using the Bhattacharyya distance.

We use this method to classify all of the textures in the CUReT database (see section 3.3), using 43 training images per class and calculating results over 100 random splits into training and test images, as in [6], giving a score of 98.2±0.1% (with $\sigma = 1$). This is at least as good as other methods using nearest-neighbour classification (see table 1), showing that a geometrically derived set of features can be at least as effective a representation for texture as cluster-centre textons calculated from the training data.

3.2. Comparison of image representations across scale

Although our local BIF-column descriptors describe the variation in local image structure across scale, they are not scale-invariant. That is, despite being able to describe interesting structures at a range of scales, each descriptor is rooted at the same 'base' scale. In order to cope with datasets which, unlike CUReT, contain significant scale-differences between images of the same class, we introduce a measure for combining distances between histograms at a range of scales.

Hayman et al. [7] successfully adopt a pure learning approach to this problem, in effect augmenting the training set with a range of artificially rescaled versions of the original training images. Although this works well on the datasets tested, it implicitly assumes that texture images have a single scale at which they are best described, which may not be the case for certain classes of textures.

We would like to maintain the links between representations of the same image analysed at different scales. For our multi-scale comparison we therefore first compute a stack of histograms for each image at a range of scales. Each histogram is calculated in exactly the same way as for our single scale implementation. The 'base scale' of the finest scale histogram – that is, the finest scale in the BIF-columns over which the histogram is computed – is $2^{-1}/_{4}$, and there is a separation between each scale of one quarter octave. For computational reasons, in this preliminary study we limit the number of

scales analysed to seven, making the base scale of the coarsest scale histogram $2^{3}/_{2}$. Thus, because our BIF-columns span four octaves, the total range of scales analysed extends from 0.84 to 22.6.

Given these histogram stacks for two images to be compared, there are two components to our comparison. The first deals with the problem of wanting to match images dependant on their structure at all scales, rather than one characteristic scale. The second addresses, for example, images taken from different distances, by allowing all possible translations of histogram stacks relative to each other, as in figure 5.

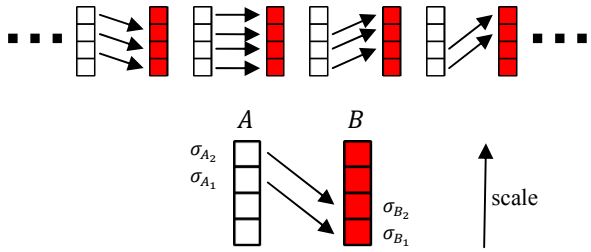

Figure 5: Multi-scale comparison of images A and B. We compare histograms computed at seven base scales, of which only four are shown here for clarity. Histogram stacks are shifted up and down in scale relative to each other to allow matching of similar features appearing at different scales in each image.

To compare stacks of normalised BIF-column histograms for images A and B, calculated at column-base scales $\sigma_{A_1}, \sigma_{A_2}, \ldots, \sigma_{A_n}$ and $\sigma_{B_1}, \sigma_{B_2}, \ldots, \sigma_{B_n}$ respectively, we take a weighted average of squared Bhattacharyya distances computed at each pair of scales $(\sigma_{A_i}, \sigma_{B_i})$,

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

where $h(I;\, \sigma_j)$ is the normalised BIF-column histogram of image I computed at 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 allows direct comparison of distances for differently shifted comparisons, allowing the multi-scale scheme to be incorporated into our nearest neighbour classifier.

We emphasize the difference between multi-scale BIF-columns, which describe the *local* variation of the texture across scale (the local deep structure) at each point in an image analysed at a certain scale; and multi-scale histogram comparison which describes the *global* variation over scale of the texture (representation) itself.

3.3. Classification results

We present classification results on three commonly used datasets, using exactly the same method on each with no tuning of parameters. To review, the main elements of this method are i) the representation of a texture at some scale as a histogram over a universal, geometrically-defined dictionary of rotationally-invariant features (BIF-columns); and ii) nearest-neighbour classification using the multi-scale measure of section 3.2, which is based on the Bhattacharyya distance.

The CUReT database [20] contains 61 classes, each consisting of 205 images of some physical texture sample photographed under a range of viewing and lighting angles but without significant variation in scale or in-plane rotation. In accordance with other studies which use CUReT for classification, we consider only the 92 images per class which allow the extraction of a 200x200 pixel foreground region of texture.

The KTH-TIPS dataset [7] expands CUReT by photographing 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 200x200 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 [6] in treating it as a stand-alone dataset.

UIUCTex [8] contains 25 classes, each of 40 images (640x480 pixels) with significant changes in scale and viewpoint as well as non-rigid deformations; although with less severe lighting variations than CUReT. Unlike the other two datasets, UIUCTex is uncalibrated. In terms of intra-class variations in appearance, this is the most challenging of the commonly used testbeds for texture classification.

For testing, we select 100 different random training/test splits for each dataset, as in [6], and report the mean number of correct classifications. Comparative results are presented in Table 1, for training on 43, 40 and 20 images per class from the CUReT, KTH-TIPS and UIUCTex datasets respectively.

First, note that our multi-scale scheme produces a small but significant improvement on the CUReT database over the single scale results reported in section 3.1. Since CUReT does not contain significant intra-class variation in scale, scale-shifting of histogram stacks (see figure 5) would not be expected to be useful. Figure 6 confirms this, showing that shifting is rarely used when CUReT images are classified correctly by our multi-scale algorithm, but is often seen in incorrect categorizations, suggesting that shifting is only used in cases where no good match is available at the same scale. This improvement therefore reflects a combination of the added descriptive power of using filters at scales better matched to the structures present in certain textures, with

appropriate use of this multi-scale information in classification. Figure 7 shows examples of images which are misclassified by our algorithm.

	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±0.1%	**98.8±0.1%**	**98.5±0.1%**
Varma & Zisserman - MR8 [4]	97.43%		
Varma & Zisserman – Joint [3]	98.03%	78.4±2.0%†	92.4±2.1%†
Hayman et al. [7]	98.46±0.09%	92.0±1.3%†	94.8±1.2%†
Lazebnik et al. [8]	72.5±0.7%†	96.03%	91.3±1.4%†
Zhang et al. [6]	95.3±0.4%	98.3±0.5%	95.5±1.3%
Broadhurst [21]	**99.22±0.34%**		

Table 1: Classification scores for our, and a number of state-of-the-art methods, on three well-known datasets. Results are as originally reported, except for those marked † which are taken from [6].

Despite not being modified to suit each dataset, our method produces consistently good results across all three databases. To the best of our knowledge, the results for multi-scale BIF-columns exceed the best reported for the UIUCTex and KTH-TIPS datasets. For CUReT, Broadhurst [21] has achieved 99.22% correct classification by using a Gaussian Bayes Classifier with marginal filter distributions. The method which we report achieves superior performance to other methods which use nearest-neighbour classification.

4. Conclusions and Further Work

We have presented a multi-scale texture classification algorithm which, without any tuning of paramaters, produces what we believe to be state-of-the-art results on two widely-used texture datasets; and close to state-of-the-art results on a third.

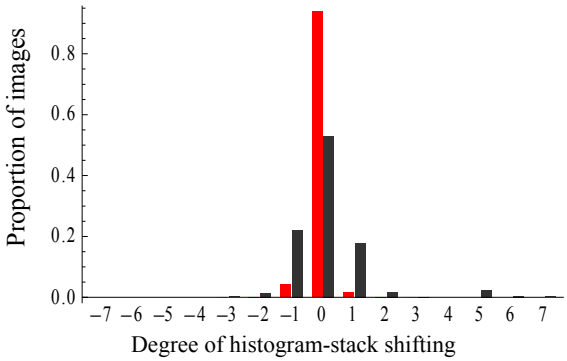

Figure 6: The distribution of histogram-stack shifts (figure 5) used by our multi-scale algorithm for images which are correctly (red) and incorrectly (black) classified.

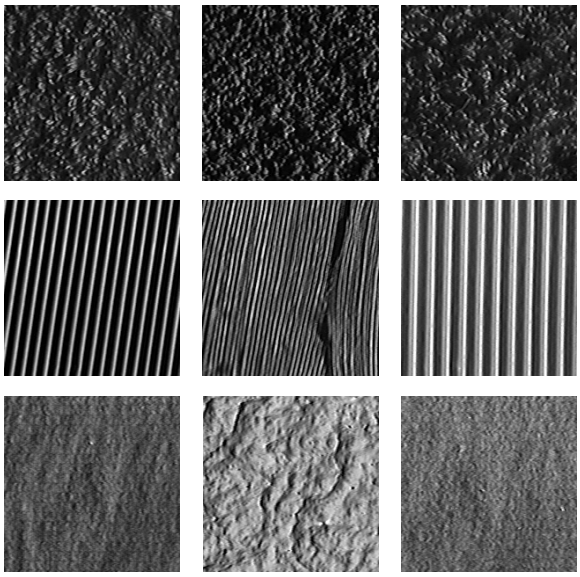

Figure 7: Three examples of texture images from the CUReT dataset [20] which are misclassified by our multi-scale BIF-columns algorithm (left); the images to which they were incorrectly matched (centre); and 'nearest misses' from the correct class (right). Top: Terrycloth mistaken for Pebbles. Like many misclassified textures, these are perceptually very similar. Middle: Ribbed Paper mistaken for Corn Husk. In this particular case, scale invariance may have been a disadvantage. Bottom: Corduroy mistaken for Plaster B, showing very little perceptual similarity.

The novelty of our system lies in the fact that we use a geometrically-derived dictionary of features over which images are represented, rather than pre-training a dictionary of textons for each dataset by clustering. This results in a simpler and more general approach.

That a universal dictionary can outperform one trained to the dataset in question shows that clustering is certainly not an optimal way to produce features for statistical texture representation.

The main focus of this paper has been on image representation and so we have avoided using more sophisticated classifiers for ease of implementation. However, a number of studies [6, 7, 22] have shown that Support Vector Machines consistently outperform nearest neighbour matching for tasks of this kind, and so this is an obvious direction for improvement.

We also make no claims about the optimality of our BIF-columns (as opposed to other ways of combining BIFs) or multi-scale scheme, both of which we plan to investigate further.

Acknowledgements

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

References

[1] T. Leung and J. Malik, "Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons," *International Journal of Computer Vision,* vol. 43, pp. 29-44, 2001.

[2] M. Varma and A. Zisserman, "Classifying Images of Materials: Achieving Viewpoint and Illumination Independence," in *Computer Vision - ECCV 2002: 7th European Conference on Computer Vision, Copenhagen, Denmark, May 28-31, 2002. Proceedings, Part III*, 2002, p. 255.

[3] M. Varma and A. Zisserman, "Texture classification: are filter banks necessary?," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 2003, pp. II-691-8 vol.2.

[4] M. Varma and A. Zisserman, "A Statistical Approach to Texture Classification from Single Images," *International Journal of Computer Vision,* vol. 62, pp. 61-81, 2005/04/25/ 2005.

[5] O. G. Cula and K. J. Dana, "Recognition methods for 3D textured surfaces," in *Proceedings of SPIE Conference on Human Vision and Electronic Imaging VI*, San Jose, 2001.

[6] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, "Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study," in *Computer Vision and Pattern Recognition Workshop, 2006 Conference on*, 2006, p. 13.

[7] E. Hayman, B. Caputo, M. Fritz, and J.-O. Eklundh, "On the Significance of Real-World Conditions for Material Classification," in *Computer Vision - ECCV 2004*, 2004, pp. 253-266.

[8] S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using local affine regions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 27, pp. 1265-1278, 2005.

[9] S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using affine-invariant regions," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 2003, pp. II-319-II-324 vol.2.

[10] M. Varma and R. Garg, "Locally Invariant Fractal Features for Statistical Texture Classification," in *Proceedings of the {IEEE} International Conference on Computer Vision, Rio de Janeiro, Brazil*, 2007.

[11] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, 1999, pp. 1150-1157 vol.2.

[12] L. D. Griffin and M. Lillholm, "Feature category systems for 2nd order local image structure induced by natural image statistics and otherwise.," in *SPIE 6492(09):1-11*, 2007.

[13] S. Konishi and A. L. Yuille, "Statistical cues for domain specific image segmentation with performance analysis," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, 2000, pp. 125-132 vol.1.

[14] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, pp. 971-987, 2002.

[15] L. D. Griffin, M. Lillholm, and M. Crosier, "Basic Image Features (BIFs) arising from analysis of Local Symmetry," in *ECCV '08 (submitted)*.

[16] L. D. Griffin, "The 2nd order local-image-structure solid," *IEEE Trans Patt Anal Mach Intell,* vol. 29, pp. 1355-1366, 2007.

[17] L. D. Griffin, "Symmetries of 1-D images," *Journal of Mathematical Imaging & Vision,* 2008.

[18] B. M. ter Haar Romeny, *Front-End Vision and Multi-Scale Image Analysis*: Kluwer Academic Publishers, 2003.

[19] N. A. Thacker, F. J. Aherne, and P. I. Rockett, "The Bhattacharyya Metric as an Absolute Similarity Measure for Frequency Coded Data.," *Kybernetika,* vol. 34, pp. 363-368, 1997.

[20] O. G. Cula and K. J. Dana, "Compact representation of bidirectional texture functions," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 2001, pp. I-1041-I-1047 vol.1.

[21] R. E. Broadhurst, "Statistical estimation of histogram variation for texture classification," in *Proc. Intl. Workshop on Texture Analysis and Synthesis* Beijing, 2005, pp. 25–30.

[22] B. Caputo, E. Hayman, and P. Mallikarjuna, "Class-specific material categorisation," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, 2005, pp. 1597-1604 Vol. 2.