A Statistical Approach To Material Classification Using Image Patch Exemplars

Manik Varma	Andrew Zisserman
Microsoft Research	Dept. of Engineering Science
Bangalore	University of Oxford
India 560 080	Oxford, UK OX1 3PJ

Abstract

In this paper, we investigate material classification from single images obtained under unknown viewpoint and illumination. It is demonstrated that materials can be classified using the joint distribution of intensity values over extremely compact neighbourhoods (starting from as small as 3×3 pixels square), and that this can outperform classification using filter banks with large support. It is also shown that the performance of filter banks is inferior to that of image patches with equivalent neighbourhoods.

We develop novel texton based representations which are suited to modelling this joint neighbourhood distribution for MRFs. The representations are learnt from training images, and then used to classify novel images (with unknown viewpoint and lighting) into texture classes. Three such representations are proposed, and their performance is assessed and compared to that of filter banks.

The power of the method is demonstrated by classifying 2806 images of all 61 materials present in the Columbia-Utrecht database. The classification performance surpasses that of recent state of the art filter bank based classifiers such as Leung and Malik (IJCV 01), Cula and Dana (IJCV 04), and Varma and Zisserman (IJCV 05). We also benchmark performance by classifying all the textures present in the UIUC, Microsoft Textile and the San Francisco outdoor datasets.

We conclude with discussions on why features based on compact neighbourhoods can correctly discriminate between textures with large global structure and why the performance of filter banks is not superior to that of the source image patches from which they were derived.

Index Terms

Material classification, 3D textures, textons, image patches, filter banks.

1. Introduction

Our objective, in this paper, is the classification of materials from their appearance in single images taken under unknown viewpoint and illumination conditions. The task is difficult as materials typically exhibit large intra-class, and small inter-class, variability (see Figure 1) and there aren't any widely applicable yet mathematically rigorous models which account for such transformations. The task is made even more challenging if no *a priori* knowledge about the imaging conditions is available.

Early interest in the texture classification problem focused on the pre-attentive discrimination of texture patterns in binary images [3], [26], [27], [38]. Later on, this evolved to the classification of textures in grey scale images with synthetic 2D variations [20], [22], [47]. This, in turn, has been superseded by the problem of classifying real world textures with 3D variations due to

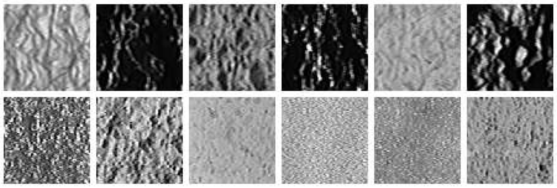

Fig. 1. Single image classification on the Columbia-Utrecht database is a demanding task. In the top row, there is a sea change in appearance (due to variation in illumination and pose) even though all the images belong to the same texture class. This illustrates large intra class variation. In the bottom row, several of the images look similar and yet belong to different texture classes. This illustrates that the database also has small inter class variation.

changes in camera pose and illumination [6], [11], [29], [31], [44], [54]. Currently, efforts are on extending the problem to the accurate classification of entire texture categories rather than of specific material instances [9], [23]. Another trend investigates how regularity information can be exploited for the analysis of near regular textures [24].

A common thread through this evolution has been the success that filter bank based methods have had in tackling the problem. As the problem has become more difficult, such methods have coped by building richer representations of filter responses. The use of large support filter banks to extract texture features at multiple scales and orientations has gained wide acceptance.

However, in this paper, we question the dominant role that filter banks have come to play in the field of texture classification. Instead of applying filter banks, we develop an alternative image patch representation based on the joint distribution of pixel intensities in a neighbourhood.

We first investigate the advantages of this image patch representation empirically. The VZ algorithm [54] gives one of the best 3D texture classification results on the Columbia-Utrecht database using the Maximum Response 8 (MR8) filters with support as large as 49×49 pixels square. We demonstrate that substituting the new patch based representation in the VZ algorithm leads to the following two results: (i) very good classification performance can be achieved using extremely compact neighbourhoods (starting from as small as 3×3) and that (ii) for any fixed size of the neighbourhood, image patches lead to superior classification as compared to filter banks with the same support. The superiority of the image patch representation is empirically demonstrated by classifying all 61 materials present in the Columbia-Utrecht database and showing that the results outperform the VZ algorithm using the MR8 filter bank. Results are also presented for the UIUC [30], San Francisco [29] and Microsoft Textile [42] databases.

We then discuss theoretical reasons as to why small image patches can correctly discriminate between textures with large global structure and also challenge the popular belief that filter bank features are superior for classification as compared to the source image patches from which they were derived. Finally, we present results on texture synthesis and denoising to reinforce the fact that the new representation can be learnt accurately even in high dimensional, image patch space. A preliminary version of this work appeared in [52].

2. Background

Texture research is generally divided into five canonical problem areas: (1) synthesis; (2) classification; (3) segmentation; (4) compression; and (5) shape from texture. The first four areas have come to be heavily influenced by the use of wavelets and filter banks, with wavelets being particularly effective at compression, while filter banks have lead the way in classification, segmentation and synthesis.

The success in these areas was largely due to learning a fuller statistical representation of filter bank responses. It was fuller in three respects: first, the filter response *distribution* was learnt (as opposed to recording just the low order moments of the distribution); second, the *joint* distribution, or co-occurrence, of filter responses was learnt (as opposed to independent distributions for each filter); and third, simply more filters were used than before to measure texture features at many scales and orientations.

These filter response distributions were learnt from training images and represented by clusters or histograms. The distributions could then be used for classification, segmentation or synthesis. For instance, classification could be achieved by comparing the distribution of a novel texture image to the model distributions learnt from the texture classes. Similarly, synthesis could be achieved by constructing a texture having the same distribution as the target texture. As such, the use of filter banks has become ubiquitous and unquestioned.

However, even though there has been ample empirical evidence to suggest that filter banks and wavelets can lead to good performance, not much rigorous theoretical justification has been provided as to their optimality or, even for that matter, their necessity for texture classification, synthesis or segmentation. In fact, the supremacy of filter banks for texture synthesis was brought into question by the approach of Efros and Leung [15]. They demonstrated that superior synthesis results could be obtained using local pixel neighbourhoods directly, without resorting to large scale filter banks. In a related development, Zalesny and Van Gool [58] also eschewed filter banks in favour of a Markov random field (MRF) model. More recently, and following the same trend, [56] showed that small patches can provide an alternative to filter banks for texture edge detection and segmentation.

Both [15], [58] put MRFs firmly back on the map as far as texture synthesis was concerned. Efros and Leung gave a computational method for generating a texture with similar MRF statistics to the original sample, but without explicitly learning or even representing these distributions. Zalesny and Van Gool, using a subset of all available cliques present in a neighbourhood, showed that it was possible to learn and sample from a parametric MRF model given enough computational power.

In this paper, it is demonstrated that the second of the canonical problems, texture classification, can also be tackled effectively by employing only local neighbourhood distributions, with representations inspired by MRF models.

2.1. The Columbia-Utrecht database

In this section, we describe the Columbia-Utrecht (CUReT) database [12] and its level of difficulty for single image classification. The database contains images of 61 materials and includes many surfaces that we might commonly see in our environment. It has textures that are rough, those which have specularities, exhibit anisotropy, are man-made and many others. The variety of textures present in the database is shown in Figure 2.

Each of the materials in the database has been imaged under 205 different viewing and illumination conditions. The effects of specularities, inter-reflections, shadowing and other surface normal variations are plainly evident and can be seen in Figure 1 where their impact is highlighted due to varying imaging conditions. This makes the database far more challenging for a classifier than the often used Brodatz collection where all such effects are absent.

While the CUReT database has now become a benchmark and is widely used to assess classification performance, it also has some limitations. These are mainly to do with the way the images have been photographed and the choice of textures. For the former, there is no significant scale change for most of the materials and very limited in-plane rotation. With regard to choice of texture, the most serious drawback is that multiple instances of the same texture are present for only a few of the materials, so intra-class variation cannot be thoroughly investigated. Hence,

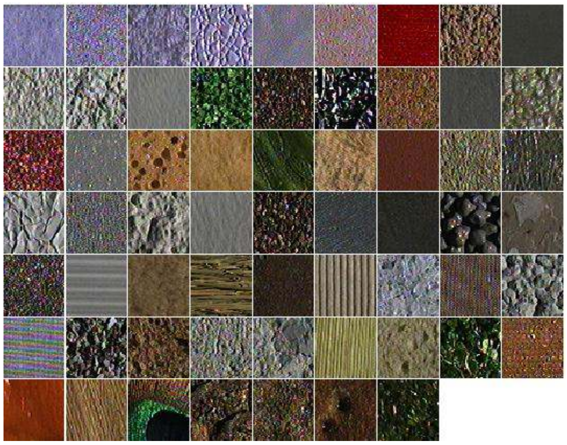

Fig. 2. One image of each of the materials present in the Columbia-Utrecht (CUReT) database. Note that all images are converted to grey scale in our classification scheme and no use of colour information is made whatsoever.

it is difficult to make generalisations. Nevertheless, it is still one of the largest and toughest databases for a texture classifier to deal with.

All 61 materials present in the database are included in the experimental setup used in Sections 4 and 5. For each material, there are 118 images where the azimuthal viewing angle is less than 60 degrees. Out of these, 92 images are chosen for which a sufficiently large region of texture is visible across all materials. The remaining images are not included as they do not have large enough foreground texture regions where large support filter banks can be applied. A central 200×200 region is cropped from each of the selected images and the remaining background discarded. The selected regions are converted to grey scale and then intensity normalised to have zero mean and unit standard deviation. Thus, no colour information is used in any of the experiments and we make ourselves invariant to affine changes in the illuminant intensity. The cropped CUReT database has a total of $61 \times 92 = 5612$ images. Out of these, 46 images per class are randomly chosen for training and the remaining 46 per class are chosen for testing. The cropped CUReT database can be downloaded from [1].

3. A review of the VZ classifier

The classification problem being tackled is the following: given an image consisting of a single texture obtained under unknown illumination and viewpoint, categorise it as belonging to one of a set of pre-learnt texture classes. Leung and Malik's influential paper [31] established much of the framework for this area – filter response textons, nearest neighbour classification using the χ^2 statistic, testing on the CUReT database, etc. Later algorithms such as the BFH classifier [11] and the VZ classifier [54] have built on this paper and extended it to classify single images without compromising accuracy. In turn, [6], [9], [23] have achieved even superior results by keeping the MR8 filter bank representation of the VZ algorithm but replacing the nearest neighbour classifier with SVMs or Gaussian-Bayes classifiers.

The VZ classifier [54] is divided into two stages: a learning stage where texture models are learnt from training examples by building statistical descriptions of filter responses, and a classification stage where novel images are classified by comparing their distributions to the learnt models.

In the learning stage, training images are convolved with a chosen filter bank to generate filter responses. These filter responses are then aggregated over images from a texture class and clustered. The resultant cluster centres form a dictionary of exemplar filter responses which are called textons. Given a texton dictionary, a model is learnt for a particular training image by labelling each of the image pixels with the texton that lies closest to it in filter response space. The model is the normalised frequency histogram of pixel texton labellings, i.e. an S-vector of texton probabilities for the image, where S is the size of the texton dictionary. Each texture class is represented by a number of models corresponding to training images of that class.

In the classification stage, the set of learnt models is used to classify a novel (test) image into one of the 61 textures classes. This proceeds as follows: the filter responses of the test image are generated and the pixels labelled with textons from the texton dictionary. Next, the normalised frequency histogram of texton labellings is computed to define an S-vector for the image. A nearest neighbour classifier is then used to assign the texture class of the nearest model to the test image. The distance between two normalised frequency histograms is measured using the χ^2 statistic, where $\chi^2(\mathbf{x}, \mathbf{y}) = \frac{1}{2}\sum_i \frac{(x_i - y_i)^2}{x_i + y_i}$.

The performance of six filter banks was contrasted in [50]. These include 4 filter banks based on the Maximum Response filter set (BFS, MR8, MR4 and MRS4), the filter bank of Schmid (S) [43] and the filter bank of Leung and Malik (LM) [31] which was also used by Cula and Dana [11]. It was demonstrated that the rotationally invariant, multi-scale, Maximum Response MR8 filter bank (described below) yields better results than any of the other filters. Hence, in this paper, we present comparisons with the MR8 filter bank.

3.1. Filter bank

The MR8 filter bank consists of 38 filters but only 8 filter responses. The filters include a Gaussian and a Laplacian of a Gaussian (LOG) filter both at scale $\sigma = 10$, an edge (first derivative) filter at 6 orientations and 3 scales and a bar (second derivative) filter also at 6 orientations and the same 3 scales (σ_x,σ_y)=$\{$(1,3), (2,6), (4,12)$\}$. The response of the isotropic filters (Gaussian and LOG) are used directly. However, in a manner somewhat similar to [41], the responses of the oriented filters (bar and edge) are "collapsed" at each scale by using only the maximum filter responses across all orientations. This gives 8 filter responses in total and ensures that the filter responses are rotationally invariant. The MR4 filter bank only employs the $(\sigma_x, \sigma_y) = (4, 12)$ scale. Another 4 dimensional variant, MRS4, achieves rotation and scale invariance by selecting the maximum response over both orientation and scale [50]. Matlab code for generating these filters, as well as the LM and S sets, is available from [2].

3.2. Pre-processing

The following pre-processing steps are applied before going ahead with any learning or classification. First, every filter in the filter bank is made mean zero. It is also L_1 normalised so that the responses of all filters lie roughly in the same range. In more detail, every filter F_i is divided by $||F_i||_1$ so that the filter has unit L_1 norm. This helps vector quantisation, when using Euclidean distances, as the scaling for each of the filter response axes becomes the same [37]. Note that dividing by $||F_i||_1$ also scale normalises [35] the Gaussians (and their derivatives) used in the filter bank.

Second, following [37] and motivated by Weber's law, the filter response at each pixel $\mathbf{x}$ is

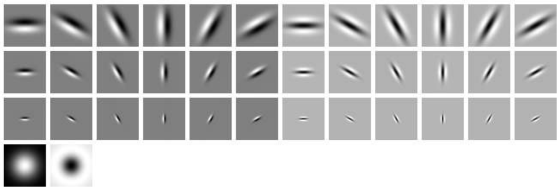

Fig. 3. The MR8 filter bank consists of 2 anisotropic filters (an edge and a bar filter, at 6 orientations and 3 scales), and 2 rotationally symmetric ones (a Gaussian and a Laplacian of Gaussian). However only 8 filter responses are recorded by taking, at each scale, the maximal response of the anisotropic filters across all orientations.

(contrast) normalised as

$$\mathbf{F}(\mathbf{x}) \leftarrow \mathbf{F}(\mathbf{x}) \left[\log\left(1 + L(\mathbf{x})/0.03\right)\right]/L(\mathbf{x}) \tag{1}$$

where $L(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|_2$ is the magnitude of the filter response vector at that pixel. This was empirically determined to lead to better classification results.

3.3. Implementation details

To learn the texton dictionary, filter responses of 13 randomly selected images per texture class (taken from the set of training images) are aggregated and clustered via the *K-Means* algorithm [14]. $K = 10$ textons are learnt from each of the 61 texture classes present in the CUReT database resulting in a dictionary comprising $61 \times 10 = 610$ textons. In previous work [54], we had explored the idea of learning *universal* texton dictionaries from a subset of the 61 texture classes. While the performance using such dictionaries was adequate, the performance obtained by learning textons from all classes was better. In addition, it was not found necessary to prune the texton dictionary or merge textons as in [31]. On the contrary, larger texton dictionaries tended to give better performance (see [53], [54] for details).

Under this setup, the VZ classifier using the MR8 filter bank achieves an accuracy rate of 96.93% while classifying all 2806 test images into 61 classes using 46 models per texture. This will henceforth be referred to as VZ Benchmark. The best results for MR8 are 97.43% obtained when a dictionary of 2440 textons is used, with 40 textons being learnt per class.

4. The image patch based classifiers

In this section, we investigate the effect of replacing filter responses with the source image patches from which they were derived. The rationale for doing so comes from the observation that convolution to generate filter responses can be rewritten as an inner product between image patch vectors and the filter bank matrix. Thus, a filter response is essentially a lower dimensional projection of an image patch onto a linear subspace spanned by the vector representation of the individual filters (obtained by row reordering each filter mask).

The VZ algorithm is now modified so that filter responses are replaced by their source image patches. Thus, the new classifier is identical to the VZ algorithm except that, at the filtering stage, instead of using a filter bank to generate filter responses at a point, the raw pixel intensities of an

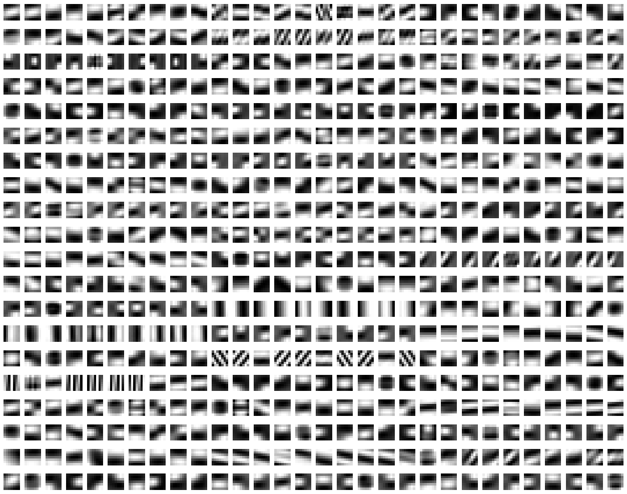

Fig. 4. Image patch textons learnt from the CUReT database using neighbourhoods of size 7 × 7.

$N \times N$ square neighbourhood around that point are taken and row reordered to form a vector in an N^2 dimensional feature space. All pre and post processing steps are retained (images are made zero mean and unit variance while patch vectors are contrast normalised using Weber's law) and no other changes are made to the classifier. Hence, in the first stage of learning, all the image patches from the selected training images in a texture class are aggregated and clustered. The set of cluster centres from all the classes comprises the texton dictionary. The textons now represent exemplar image patches rather than exemplar filter responses (see Figure 4). However, the model corresponding to a training image continues to be the histogram of texton frequencies, and novel image classification is still achieved by nearest neighbour matching using the χ^2 statistic. This classifier will be referred to as the Joint classifier. Figure 5 highlights the main difference in approach between the Joint classifier and the MR8 based VZ classifier.

We also design two variants of the Joint classifier – the Neighbourhood classifier and the MRF classifier. Both of these are motivated by the recognition that textures can often be considered realisations of a Markov random field. In an MRF framework [18], [33], the probability of the central pixel depends only on its neighbourhood. Formally,

$$p(I(\mathbf{x}_c)|I(\mathbf{x}), \forall \mathbf{x} \neq \mathbf{x}_c) = p(I(\mathbf{x}_c)|I(\mathbf{x}), \forall \mathbf{x} \in \mathcal{N}(\mathbf{x}_c)) \tag{2}$$

where $\mathbf{x}_c$ is a site in the 2D integer lattice on which the image I has been defined and $\mathcal{N}(\mathbf{x}_c)$ is the neighbourhood of that site. In our case, $\mathcal{N}$ is defined to be the $N \times N$ square neighbourhood

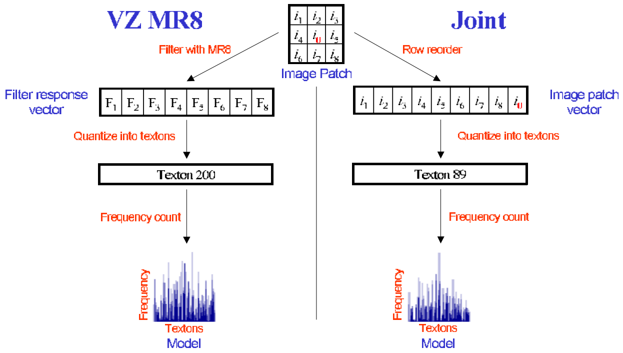

Fig. 5. The only difference between the Joint and the VZ MR8 representations is that the source image patches are used directly in the Joint representation as opposed to the derived filter responses in VZ MR8.

(excluding the central pixel). Thus, although the value of the central pixel is significant, its distribution is conditioned on its neighbours alone. The Neighbourhood and MRF classifiers are designed to test how significant this conditional probability distribution is for classification.

For the Neighbourhood classifier, the central pixel is discarded and only the neighbourhood is used for classification. Thus, the Neighbourhood classifier is essentially the Joint classifier retrained on feature vectors drawn only from the set of $\mathcal{N}$: i.e. the set of $N \times N$ image patches with the central pixel left out. For example, in the case of a 3×3 image patch, only the 8 neighbours of every central pixel are used to form feature vectors and textons.

For the MRF classifier we go to the other extreme and, instead of ignoring the central pixel, explicitly model $p(I(\mathbf{x}_c), I(\mathcal{N}(\mathbf{x}_c)))$, i.e. the joint distribution of the central pixels and its neighbours. Up to now, textons have been used to implicitly represent this joint PDF. The representation is implicit because, once the texton frequency histogram has been formed, neither the probability of the central pixel nor the probability of the neighbourhood can be recovered straightforwardly by summing (marginalising) over the appropriate textons. Thus, the texton representation is modified slightly so as to make explicit the central pixel's PDF within the joint and to represent it at a finer resolution than its neighbours (in the Neighbourhood classifier, the central pixel PDF was discarded by representing it at a much coarser resolution using one bin).

To learn the PDF representing the MRF model for a given training image, the neighbours' PDF is first represented by textons as was done for the Neighbourhood classifier – i.e. all pixels but the central are used to form feature vectors in an $N^2 - 1$ dimensional space which are then labelled

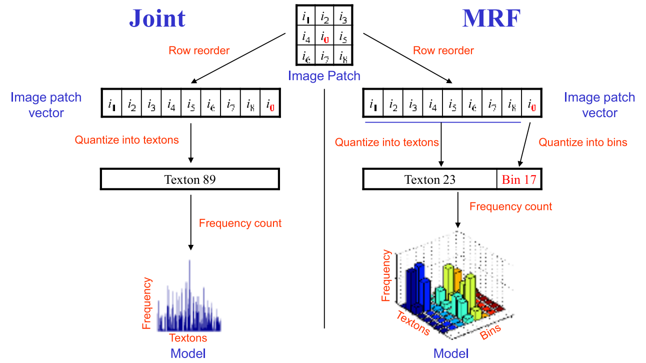

Fig. 6. MRF texture models as compared to those learnt using the Joint representation. The only point of difference is that the central pixel PDF is made explicit and stored at a higher resolution. The Neighbourhood representation can be obtained from the MRF representation by ignoring the central pixel.

using the same dictionary of 610 textons. Then, for each of the S_N textons in turn ($S_N = 610$ is the size of the neighbourhood texton dictionary), a one dimensional distribution of the central pixels' intensity is learnt and represented by an S_C bin histogram. Thus the representation of the joint PDF is now an $S_N \times S_C$ matrix. Each row is the PDF of the central pixel for a given neighbourhood intensity configuration as represented by a specific texton. Figure 6 highlights the differences between MRF models and models learnt using the Joint representation. Using this matrix, a novel image is classified by comparing its MRF distribution to the learnt model MRF distributions by computing the χ^2 statistic over all elements of the $S_N \times S_C$ matrix.

$N \times N$	Joint Classifier (%)	Neighbourhood Classifier (%)	MRF with 90 bins (%)
3×3	95.33	94.90	95.87
5×5	95.62	95.97	97.22
7×7	96.19	96.08	97.47
	(a)	(b)	(c)

TABLE I. Comparison of classification results of all 61 textures in the CUReT database for different $N \times N$ neighbourhood (patch) sizes: (a) all the pixels in an image patch are used to form vectors in an N^2 feature space; (b) all but the central pixel are used (i.e. an $N^2 - 1$ space); (c) the MRF classifier where 90 bins are used to represent the joint neighbourhood and central pixel PDF. A dictionary of 610 textons learnt from all 61 textures is used throughout. Notice that the performance using these small patches is as good as that achieved by the multi orientation, multi scale, MR8 filter bank with 49×49 support (96.93% using 610 textons and 97.43% using 2440 textons).

Table I presents a comparison of the performance of the Joint, Neighbourhood and MRF classifiers when tested on the CUReT database (see Section 2.1 for experimental setup details). Image patches of size 3×3, 5×5 and 7×7 are tried while using a dictionary of 610 textons. For the Joint classifier, it is remarkable to note that classification results of over 95% are achieved using patches as small as 3×3. In fact, the classification result for the 3×3 neighbourhood is actually better than the results obtained by using the MR4 (91.70%), MRS4 (94.23%), LM (94.65%) or S (95.22%) filter banks. This is strong evidence that there is sufficient information in the joint distribution of the nine intensity values (the central pixel and its eight neighbours) to discriminate between the texture classes. For the Neighbourhood classifier, as shown in column (b), there is almost no significant variation in classification performance as compared to using all the pixels in an image patch. Classification rates for $N = 5$ are slightly better when the central pixel is left out and marginally poorer for the cases of $N = 3$ and $N = 7$. Thus, the joint distribution of the neighbours is largely sufficient for classification. Column (c) presents a comparison of the performance of the Joint and Neighbourhood classifiers to the MRF classifier when a resolution of 90 bins is used to store the central pixels' PDF. As can be seen, the MRF classifier does better than both the Joint and Neighbourhood classifiers. What is also interesting is that the performance of the MRF classifier using 7×7 patches (97.47%) is at least as good as the best performance achieved by the multi-scale MR8 filter bank with support 49×49 (97.43% using 2440 textons).

This result showing that image patches can outperform filters raises the important question of whether filter banks are actually providing beneficial information for classification, for example perhaps by increasing the signal to noise ratio, or by extracting useful features. We first address this issue experimentally, by determining the classification performance of filter banks across many different parameter settings and seeing if performance is ever superior to equivalent patches.

In order to do so, the performance of the VZ classifier using the MR8 filter bank (VZ MR8) is compared to that of the Joint, Neighbourhood and MRF classifiers as the size of the neighbourhood is varied. In each experiment, the MR8 filter bank is scaled down so that the support of the largest filters is the same as the neighbourhood size. Once again, we emphasise that the MR8 filter bank is chosen as its performance is better than all the other filter banks studied. Figure 7 plots the classification results. It is apparent that for any given size of the neighbourhood, the performance of VZ MR8 using 610 textons is worse than that of the Joint

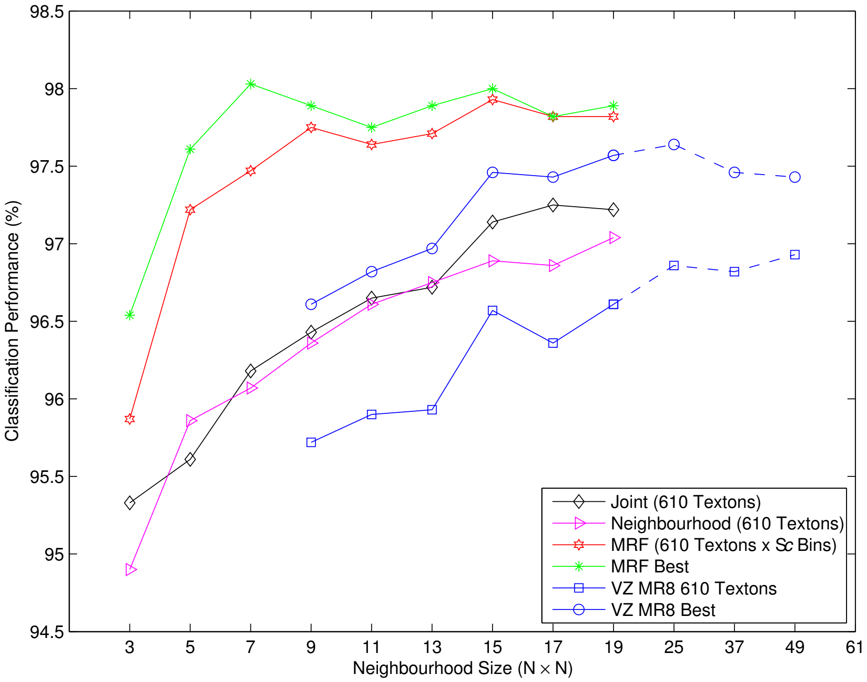

Fig. 7. Classification results as a function of neighbourhood size: The MRF Best curve shows results obtained for the best combination of texton dictionary and number of bins for a particular neighbourhood size. For neighbourhoods up to 11×11, dictionaries of up to 3050 textons and up to 200 bins are tried. For 13×13 and larger neighbourhoods, the maximum size of the texton dictionary is restricted to 1220 because of computational expense. Similarly, the VZ MR8 Best curve shows the best results obtained by varying the size of the texton dictionary. However, in this case, dictionaries of up to 3050 textons are tried for all neighbourhoods. The best result achieved by the MRF classifiers is 98.03% using a 7×7 neighbourhood with 2440 textons and 90 bins. The best result for MR8 is 97.64% for a 25×25 neighbourhood and 2440 textons. The performance of the VZ algorithm using the MR8 filter bank (VZ MR8) is always worse than any other comparable classifier at the same neighbourhood size. VZ MR8 Best is inferior to the MRF curves, while VZ MR8 with 610 textons is inferior to the Joint and Neighbourhood classifiers also with 610 textons.

or even the Neighbourhood classifiers also using 610 textons. Similarly, VZ MR8 Best is always inferior not just to MRF Best but also to MRF. To assess statistical significance, we repeat the experiment for VZ MR8 610 and Joint 610 over a thousand random partitionings of the training and test set. The results are given in Table II and show that for each neighbourhood size, and in each of the thousand random partitioning, the Joint classifier outperforms VZ MR8. The results are statistically significant since the p-value was always zero. This would suggest that using all the information present in an image patch is more beneficial for classification than relying on lower dimensional responses of a pre-selected filter bank. A classifier which is able to learn

	$9 \times 9(\%)$	$11 \times 11(\%)$	$13 \times 13(\%)$	$15 \times 15(\%)$	$17 \times 17(\%)$	$19 \times 19(\%)$
VZ MR8 610	95.06 ± 0.41	95.57 ± 0.38	95.92 ± 0.37	96.16 ± 0.37	96.30 ± 0.37	96.37 ± 0.36
Joint 610	96.38 ± 0.35	96.58 ± 0.34	96.63 ± 0.35	96.89 ± 0.33	97.11 ± 0.32	97.17 ± 0.32

TABLE II. Statistical significance: The mean and standard deviation for the VZ MR8 classifier with 610 textons and Joint classifier also with 610 textons are reported as a function of the neighbourhood size. In each case, results are reported over a thousand random partitionings of the training and test set. The performance of the Joint classifier is better than that of VZ MR8 for every one of the thousand splits for each neighbourhood size. This resulted in the p-value being zero in each case indicating that the results are statistically significant.

from all the pixel values is superior.

These results demonstrate that a classification scheme based on MRF local neighbourhood distributions can achieve very high classification rates and can outperform methods which adopt large scale filter banks to extract features and reduce dimensionality. Before turning to discuss theoretical reasons as to why this might be the case, we first explore how issues such as rotation and scale impact the image patch classifiers.

5. Scale, rotation and other datasets

Three main criticisms can be levelled at the classifiers developed in the previous section. First, it could be argued that the lack of significant scale change in the CUReT textures might be the reason why image patch based classification outperforms the multi-scale MR8 filter bank. Second, the image patch representation has a major disadvantage in that it is not rotationally invariant. Third, the reason why small image patches do so well could be because of some quirk of the CUReT dataset and that classification using small patches will not generalise to other databases. In this section, each of these three issues is addressed experimentally and it is shown that the image patch representation is as robust to scale changes as MR8, can be made rotationally invariant and generalises well to other datasets.

5.1. The effect of scale changes

To test the hypothesis that the image patch representation will not do as well as the filter bank representation in the presence of scale changes, four texture classes were selected from the CUReT database (material numbers 2, 11, 12 and 14) for which additional scaled data is available (as material numbers 29, 30, 31 and 32). The materials are shown in Figure 8.

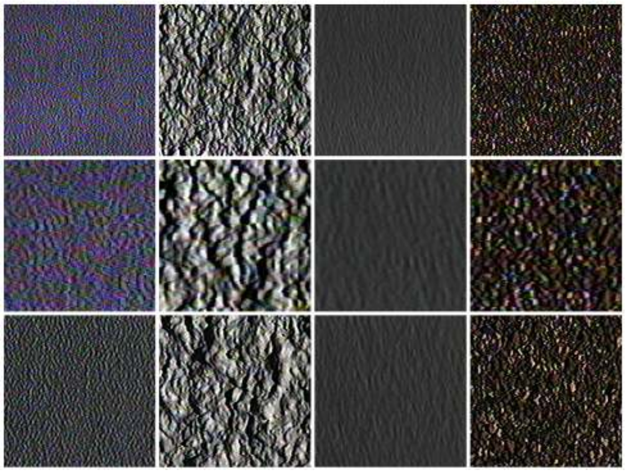

Fig. 8. The top row shows one image each from material numbers 2, 11, 12 and 14 from the CUReT database. The middle row shows the same textures scaled synthetically by a factor of two while the bottom row shows the textures scaled naturally (as material numbers 29, 30, 31 and 32).

Two experiments were performed. In the first, models were learnt only from the training images of the original textures while the test images of both the original and scaled textures were classified. In the second experiment, both test sets were classified once more but this time models were learnt from the original as well as the scaled textures. Table III shows the results of the experiments. It also tabulates the results when the experiments are repeated but this time with the images being scaled synthetically by a factor of two.

In the naturally scaled case, when classifying both texture types using models learnt only from the original textures, the MRF classifier achieves 93.48% while VZ MR8 (which contains filters at three scales) reaches only 81.25%. This shows that the MRF classifier is not being

	Naturally Scaled		Synthetically Scaled $\times 2$	
	Original (%)	Original + Scaled (%)	Original (%)	Original + Scaled (%)
MRF	93.48	100	65.22	99.73
MR8	81.25	99.46	62.77	99.73

TABLE III. Comparison of classification results of the MRF and VZ MR8 classifiers for scaled data. Models are learnt either from the original textures only or the original + scaled textures while classifying both texture types. In each case, the performance of the MRF classifier is at least as good as that using the multi-scale MR8 filter bank.

adversely affected by the scale variations. When images from the scaled textures are included in the training set as well, the accuracy rates go up 100% and 99.46% respectively. A similar trend is seen in the case when the scaled textures are generated synthetically. Both these results show that image patches cope as well with scale changes as the MR8 filter bank. We return to this issue in Section 5.3.3 when classifying the images in the UIUC database.

5.2. Incorporating rotational invariance

The fact that the image patch representation developed so far is not rotationally invariant can be a serious limitation. However, it is straight forward to incorporate invariance into the representation. There are several possibilities: (i) find the dominant orientation of the patch (as is done in the MR filters), and measure the neighbourhood relative to this orientation; (ii) marginalise the intensities weighted by the orientation distribution over angle; (iii) add rotated patches to the training set so as to make the learnt decision boundaries rotation invariant [46]; etc. In this paper, we implement option (i), and instead of using an $N \times N$ square patch, the neighbourhood is redefined to be circular with a given radius. Table IV lists the results for the Neighbourhood and MRF classifiers using circular neighbourhoods with radius 3 pixels (corresponding to a 7×7 patch) and 4 pixels (9×9 patch).

	Neighbourhood Classifier		MRF Classifier	
	Rotationally	Not	Rotationally	Not
	Invariant (%)	Invariant (%)	Invariant (%)	Invariant (%)
7×7	96.36	96.08	97.07	97.47
9×9	96.47	96.36	97.25	97.75

TABLE IV. Comparison of classification results of the Neighbourhood and MRF classifiers using the standard and the rotationally invariant image patch representations.

Using the rotationally invariant representation, the Neighbourhood classifier with a dictionary of 610 textons achieves 96.36% for a radius of 3 pixels and 96.47% for a radius of 4 pixels. This is slightly better than that achieved by the same classifier using the standard (non invariant) representation with corresponding 7×7 and 9×9 patches. The rates for the rotationally invariant MRF classifier are 97.07% and 97.25% using 610 textons and 45 bins. These results are slightly worse than those obtained using the standard representation. However, the fact that such high classification percentages are obtained strongly indicates that rotation invariance can be successfully incorporated into the image patch representation.

5.3. Results on other datasets

We now show that image patches can also be used to successfully classify textures other than those present in the CUReT database. It is demonstrated that the Joint classifier with patches of size $3 \times 3, 5 \times 5$ and 7×7 is sufficient for classifying the Microsoft Textile [42] and San Francisco [29] databases. For the UIUC database [30], while 9×9 patches already yield good results, the best results are obtained by patches of size 17×17. While the MRF classifier leads to the best results in general, we show that on these databases the Joint classifier already achieves very high performance.

5.3.1) The Microsoft Textile database: This has 16 folded materials with 20 images available of each taken under diffuse artificial lighting (see Figure 9 for an example). The impact of non-Lambertian effects is plainly visible (as it is in the Columbia-Utrecht database). Furthermore, the variations in pose and the deformations of the textured surface make it an interesting database to analyse.

For this database, the experimental setup is kept identical to the original setup of the authors. Fifteen images were randomly selected from each of the sixteen texture classes to form the training set. While all the training images were used to form models, textons were learnt from only 3 images per texture class. Various sizes of the texton dictionary $S = 16 \times K$ were tried, with $K = 10, \ldots, 40$ textons learnt per textile. The test set comprised a total of 80 images. Table V shows the variation in performance of the Joint classifier with neighbourhood size N and texton dictionary size S.

	Size of Texton Dictionary S			
$N \times N$	160 (%)	320 (%)	480 (%)	640 (%)
3×3	96.82	96.82	96.82	96.82
5×5	99.21	99.21	99.21	99.21
7×7	96.03	97.62	96.82	97.62

TABLE V. The Joint classifier performs excellently on the Microsoft Textile database – only a single image is misclassified using 5×5 patches. These results reinforce the fact that very small patches can be used to classify textures with global structure far larger than the neighbourhoods used (the image resolutions are 1024×768).

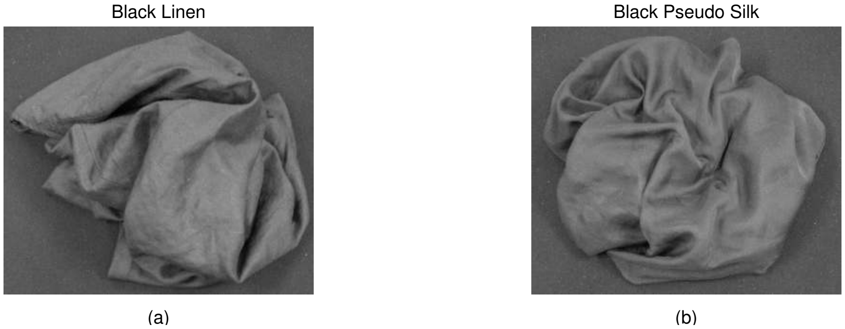

Fig. 9. Only a single image in the Microsoft Textile database is misclassified by the Joint classifier using 5×5 patches: (a) is an example of Black Linen but is incorrectly classified as Black Pseudo Silk (b).

As can be seen, excellent results are obtained using very small neighbourhoods. In fact, only a single image is misclassified using 5×5 patches (see Figure 9). These results reinforce the fact that very small patches can be used to classify textures with global structure far larger than the neighbourhoods used (the image resolutions are 1024×768).

5.3.2) The San Francisco database: This database has 37 images of outdoor scenes taken on the streets of San Francisco. Konishi and Yuille have segmented the images by hand [29] into 6 classes: Air, Building, Car, Road, Vegetation and Trunk. We work with the given segmentations and our goal is to classify each of the regions selected by Konishi and Yuille. Note that since each image has multiple texture regions present in it, the global image mean is not subtracted as was done in previous cases.

A single image is chosen for training the Joint classifier. Figure 10 shows the selected training image and its associated hand segmented regions. All the rest of the 36 images are kept as the test set. Performance is measured by the proportion of pixels that are labelled correctly during

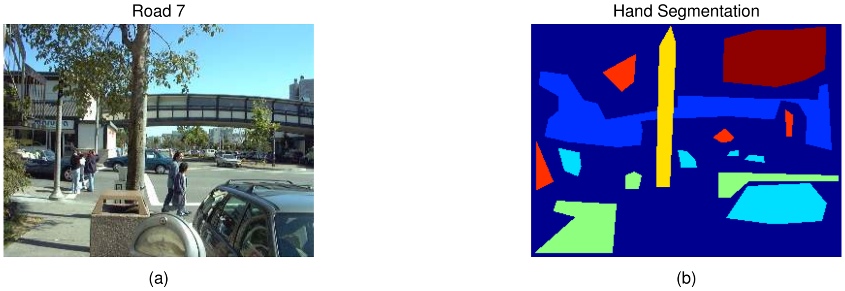

Fig. 10. The single image used for training on the San Francisco database and the associated hand segmented regions.

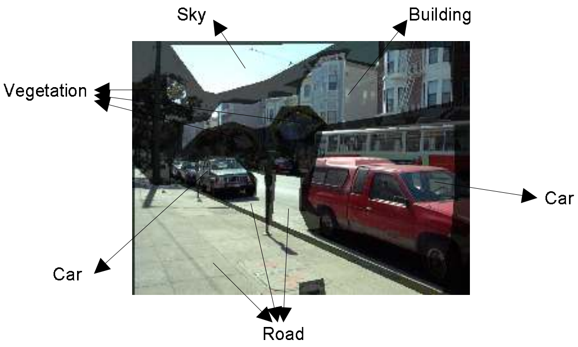

Fig. 11. Region classification results using the Joint classifier with 7×7 patches for a sample test image from the San Francisco database.

classification of the hand segmented regions. Using this setup, the Joint classifier achieves an accuracy rate of 97.9%, i.e. almost all the pixels are labelled correctly in the 36 test images. Figure 11 shows an example of a test image and the regions that were classified in it. This result again validates the fact that small image patches can be used to successfully classify textured images. In fact, using small patches is particularly appealing for databases such as the San Francisco set because large scale filter banks will have problems near region boundaries and will also not be able to produce many measurements for small, or irregularly shaped, regions.

5.3.3) The UIUC database: The UIUC texture database [30] has 25 classes and 40 images per class. The database contains materials imaged under significant viewpoint variations. Figure 12 shows examples of the materials in the database and also highlights the extreme scale and viewpoint changes.

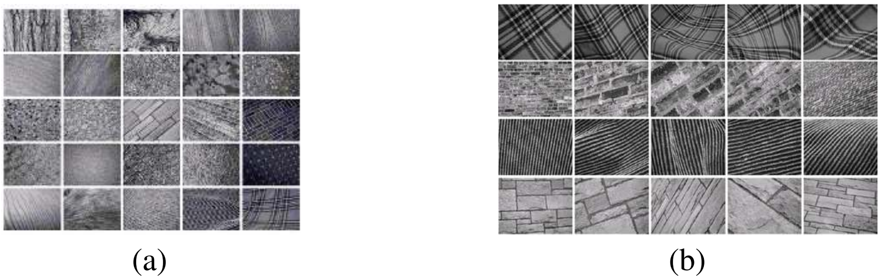

Fig. 12. The UIUC database: (a) one image from each texture class and (b) sample images from 4 texture classes showing large viewpoint variations.

We compare the performance of the Joint classifier to the performance of the rotation invariant MR8, and rotation and scale invariant MRS4, filter banks. We also compare results to the bi-Lipschitz and affine invariant state-of-the-art, nearest neighbour methods of Lazebnik *et al.* (LSP) [30], Xu *et al.* (XJF) [57] and Varma and Garg (VG) [51]. For the Joint classifier, we adopt the rotationally invariant patch based representation developed in Section 5.2. While circular patches of radius 4 already give good results, the best results are obtained by patches of radius 8. The best filter bank based results are obtained using filters of support 49×49. Texton dictionaries of size 2500 are used for all the classifiers. To assesses classification performance, M training images are randomly chosen per class while the remaining $40 - M$ images per class are taken to form the test set. Table VI presents classification results averaged over a thousand random splits of the training and test sets (the results for the LSP, XJF and VG methods are taken from Table 1 of [51]).

M	Joint (%)	MR8 (%)	MRS4 (%)	VG [51] (%)	LSP [30] (%)	XJF [57] (%)
20	97.83±0.66	92.94±1.06	90.29±1.26	95.40±0.92	93.62±0.97	93.04
15	96.94±0.77	91.16±1.11	88.47±1.25	94.09±0.98	92.42±0.99	91.11
10	95.18±0.94	88.29±1.32	85.43±1.34	91.64±1.18	90.17±1.11	88.79
05	90.17±1.44	81.12±1.74	78.44±1.77	85.35±1.69	84.77±1.54	82.99

TABLE VI. UIUC results as the number of training images M is varied. Means and standard deviations have been computed over 1000 random splits of the training and test set.

The performance of the Joint classifier is significantly superior (with p-value 0 in all experiments) to that of MR8 and MRS4. The performance gap increases as fewer and fewer images are used for training. This runs contrary to traditional expectation and bolsters the claim that the patch based representation is not necessarily adversely affected by large scale variations as compared to multi-scale filter banks. Surprisingly, the performance of the Joint classifier is also superior to the state-of-the-art bi-Lipschitz and affine invariant classifiers of [30], [51], [57].

6. WHY DOES PATCH BASED CLASSIFICATION WORK?

The results of the previous sections have demonstrated two things. First, neighbourhoods as small as 3×3 can lead to very good classification results even for textures whose global structure is far larger than the local neighbourhoods used. Second, classification using image patches is superior to that using filter banks with equivalent support. In this section, we discuss some of the theoretical reasons as to why these results might hold.

6.1. Classification using small patches

The results on the CUReT, San Francisco and Microsoft Textile databases show that small image patches contain sufficient information to discriminate between different textures. One explanation for this is illustrated in Figure 13. In (a), three images are selected from the Limestone and Ribbed Paper classes of the CUReT dataset, and scatter plots of their grey level co-occurrence matrix shown for the displacement vector $(2, 2)$ (i.e. the joint distribution of the top left and bottom right pixel in every 3×3 patch). Notice how the distributions of the two images of Ribbed Paper can easily be associated with each other and distinguished from the distribution of the Limestone image. Another example in (b) shows the same trend. Thus, 3×3 neighbourhood distributions can contain sufficient information for successful discrimination.

To take a more analytic example, consider two functions $f(x) = A \sin(\omega_f t + \delta)$ and $g(x) = A \sin(\omega_g t + \delta)$, where ω_f and ω_g are small so that f and g have large structure. Even though f and g are very similar (they are essentially the same function at different scales) it will be seen that they are easily distinguished by the Joint classifier using only two point neighbourhoods.

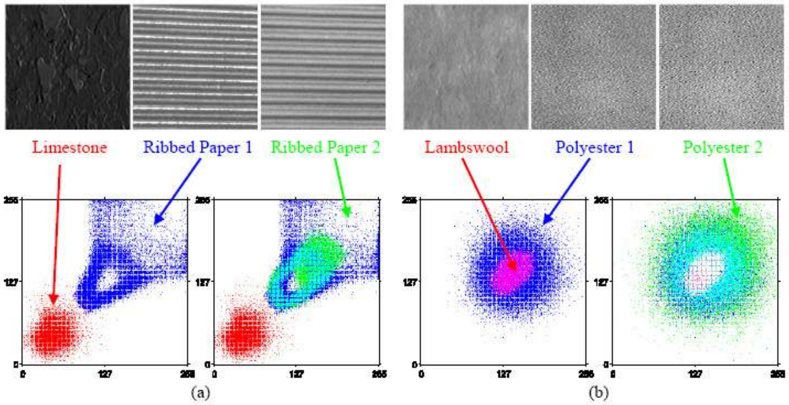

Fig. 13. Information present in 3×3 neighbourhoods is sufficient to distinguish between textures: (a) The top row shows three images drawn from two texture classes, Limestone and Ribbed Paper. The bottom row shows scatter plots of $I(\mathbf{x})$ against $I(\mathbf{x} + (2, 2))$. On the left are the distributions for Limestone and Ribbed Paper 1 while on the right are the distributions for all three images. The Limestone and Ribbed Paper distributions can easily be distinguished and hence the textures can be discriminated from this information alone. Another example is shown in (b) with the same notation.

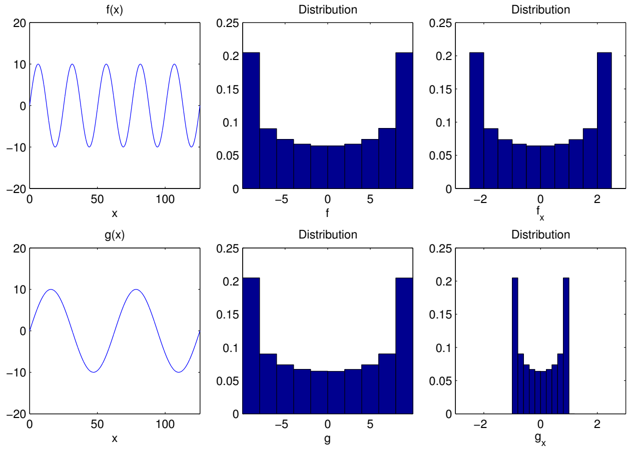

Fig. 14. Similar large scale periodic functions can be classified using the distribution of their derivatives computed from two point neighbourhoods.

Figure 14 illustrates that while the intensity distributions of f and g are identical, the distributions of their derivatives, f_x and g_x, are not. Since derivatives can be computed using just two points, these functions can be distinguished by looking at two point neighbourhoods alone.

In a similar fashion, other complicated functions such as triangular and saw tooth waves can be distinguished using compact neighbourhoods. Furthermore, the Taylor series expansion of a polynomial of degree $2N-1$ immediately shows that a $[-N, +N]$ neighbourhood contains enough information to *determine* the value of the central pixel. Thus, any function which can be locally approximated by a cubic polynomial can actually be synthesised using a $[-2, 2]$ neighbourhood. Since, in general, synthesis requires much more information than classification it is therefore expected that more complicated functions can still be distinguished just by looking at small neighbourhoods. This illustrates why it is possible to classify very large scale textures using small patches.

There also exist entire classes of textures which can not be distinguished on the basis of local information alone. One such class comprises of textures made up of the same textons and with identical first order texton statistics, but which differ in their higher order statistics. To take a simple example, consider texture classes generated by the repeated tiling of two textons (a circle and a square for instance) with sufficient spacing in between so that there is no overlap between textons in any given neighbourhood. Then, any two texture classes which differ in their tiling pattern but have identical frequencies of occurrence of the textons will not be distinguished on the basis of local information alone. However, the fact that classification rates of nearly 98% have been achieved using extremely compact neighbourhoods on three separate data sets indicates that real textures are not as simplistic as this.

The arguments in this subsection indicate that small patches might be effective at texture classification. The arguments do not imply that the performance of small patches is superior to that of arbitrarily large filter banks. However, in the next subsection, arguments are presented as to why filter banks are not superior to equivalent sized patches.

6.2. Filter banks are not superior to image patches

We now turn to the question of why filter banks do not provide superior classification as compared to their source image patches. To fix the notation, $\mathbf{f}_+$ and $\mathbf{f}_-$ will be used to denote filter response vectors generated by projecting $N \times N$ image patches $\mathbf{i}_+$ and $\mathbf{i}_-$, of dimension $d = N^2$, onto a lower dimension N_f using the filter bank $\mathbf{F}$. Thus,

$$\mathbf{f}_{\pm_{N_f \times 1}} = \mathbf{F}_{N_f \times d} \; \mathbf{i}_{\pm_{d \times 1}} \tag{3}$$

In the following discussion, we will focus on the properties of linear (including complex) filter banks. This is not a severe limitation as most popular filters and wavelets tend to be linear. Non linear filters can also generally be decomposed into a linear filtering step followed by non linear post-processing. Furthermore, since one of the main arguments in favour of filtering comes from dimensionality reduction, it will be assumed that $N_f < d$, i.e. the number of filters must be less than the dimensionality of the source image patch. Finally, it should be clarified that throughout the discussion, performance will be measured by classification accuracy rather than the speed with which classification is carried out. While the time complexity of an algorithm is certainly an important factor and can be critical for certain applications, our focus here is on achieving the best possible classification results.

The main motivations which have underpinned filtering (other than biological plausibility) are: (i) dimensionality reduction, (ii) feature extraction at multiple scales and orientations, and (iii) noise reduction and invariance. Arguments from each of these areas are now examined to see whether filter banks can lead to better performance than image patches.

6.2.1) Dimensionality Reduction: Two arguments have been used from dimensionality reduction. The first, which comes from optimal filtering, is that an optimal filter can increase the separability between key filter responses from different classes and is therefore beneficial for classification [25], [40], [49]. The second argument, from statistical machine learning, is that reducing the dimensionality is desirable because of better parameter estimation (improved clustering) and also due to regularisation effects which smooth out noisy filter responses and prevent over-fitting [5], [8], [14], [21]. We examine both arguments in turn to see whether such factors can compensate for the inherent loss of information associated with dimensionality reduction. For a more comprehensive discussion of these issues please refer to [5], [45].

6.2.1.1) Increasing separability: Since convolution with a linear filter is equivalent to linearly projecting onto a lower dimensional space, the choice of projection direction determines the distance between the filter responses. Suppose we have two image patches $\mathbf{i}_{\pm}$, with filter responses $\mathbf{f}_{\pm}$ computed by orthogonal projection as $\mathbf{f}_{\pm} = \mathbf{F}i_{\pm}$. Then the distance between $\mathbf{f}_{+}$ and $\mathbf{f}_{-}$ is clearly less than the distance between $\mathbf{i}_{+}$ and $\mathbf{i}_{-}$ (where the rows of $\mathbf{F}$ span the hyperplane orthogonal to the projection direction). The choice of $\mathbf{F}$ affects the separation between $\mathbf{f}_{+}$ and $\mathbf{f}_{-}$, and the optimum filter maximises it, in the manner of a Fisher Linear Discriminant, but the scaled distance between the projected points cannot exceed the original. This holds true for many popular distance measures including the Euclidean, Mahalanobis and the signed perpendicular distance [4] (analogous results hold when $\mathbf{F}$ is not orthogonal). It is also well known [28] that under Bayesian classification, the Bayes error either increases or remains at least as great when the dimensionality of a problem is reduced by linear projection. However, the fact that the Bayes error has increased for the low dimensional filter responses does not mean the classification is necessarily worse. This is because of issues related to noise and over-fitting which brings us to the second argument from dimensionality reduction for the superiority of filter banks.

6.2.1.2) Improved parameter estimation: The most compelling argument for the use of filters comes from statistical machine learning where it has often been noted that dimensionality reduction can lead to fewer training samples being needed for improved parameter estimation (better clustering) and can also regularise noisy data and thereby prevent over-fitting. The assumptions underlying these claims are that textures occupy a low dimensional subspace of image patch space and if the patches could be projected onto this true subspace (using a filter bank) then the dimensionality of the problem would be reduced without resulting in any

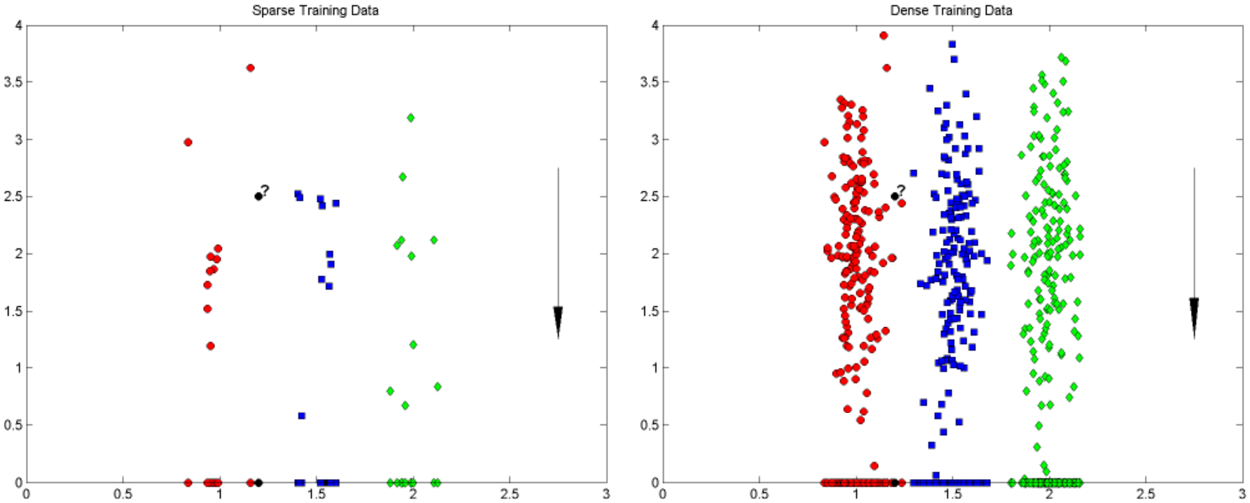

Fig. 15. Projecting the data onto lower dimensions can have a beneficial effect when not much training data is available. A nearest neighbour classifier misclassifies a novel point in the original, high dimensional space but classifies it correctly when projected onto the x axis. This problem is mitigated when there is a lot of training data available. Note that it is often not possible to know *a priori* the correct projection directions. If it were, then misclassifications in the original, high dimensional space can be avoided by incorporating such knowledge into the distance function. Indeed, this can even lead to superior classification unless *all* the information along the remaining dimensions is noise.

information loss. This would be particularly beneficial in cases where only a limited amount of training data is available as the higher dimensional patch representation would be prone to over-fitting (see Figure 15).

While these are undoubtedly sound claims there are three reasons why they might not lead to the best possible classification results. The first is due to the great difficulty associated with identifying a texture's true subspace (in a sense, this itself is one of the holy grails of texture analysis). More often than not, only approximations to this true subspace can be made and these result in a frequent loss of information when projecting downwards.

The second counter argument comes from the recent successes of boosting [48] and kernel methods [45]. Dimensionality reduction is necessary if one wants to accurately model the true texture PDF. However, both boosting and kernel methods have demonstrated that for classification purposes a better solution is to actually project the data non-linearly into an even higher (possibly infinite) dimensional space where the separability between classes is increased. Thus the emphasis is on maximising the distance between the classes and the decision boundary rather than trying to accurately model the true texture PDF (which, though ideal, is impractical). In particular, the kernel trick, when implemented properly, can lead to both improved classification and generalisation without much associated overhead and with none of the associated losses of downward projection. The reason this argument is applicable in our case is because it can be shown that χ^2, with some minor modifications, can be thought of as a Mercer kernel [55]. Thus, the patch based classifiers take the distribution of image patches and project it into the much higher dimensional χ^2 space where classification is carried out. The filter bank based VZ algorithm does the same but it first projects the patches onto a lower dimensional space which results in a loss of information. This is the reason why the performance of filter banks, such as MR8, is consistently inferior to their source patches.

The third argument is an engineering one. While it is true that clustering is better and that parameters are estimated more accurately in lower dimensional spaces, Domingos and Pazzani [13] have shown that even gross errors in parameter estimation can have very little effect on classification. This is illustrated in Figure 16 which shows that even though the means and covariance matrices of the true likelihood are estimated incorrectly, 98.6% of the data is still correctly classified, as the probability of observing the data in much of the incorrectly classified regions is vanishingly small.

Another interesting result, which supports the view that accurate parameter estimation is not necessary for accurate classification, is obtained by selecting the texton dictionary at random

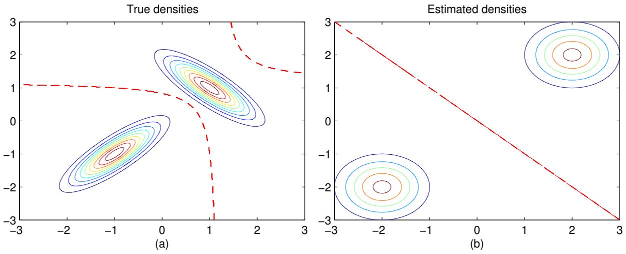

Fig. 16. Incorrect parameter estimation can still lead to good classification results: the true class conditional densities of two classes (defined to be Gaussians) are shown in (a) along with the MAP decision boundary obtained using equal priors (dashed red curves). In (b) the estimated likelihoods have gross errors. The estimated means have relative errors of 100% and the covariances are estimated as being diagonal leading to a very different decision boundary. Nevertheless the probability of misclassification (computed using the true Gaussian distributions for the probability of occurrence, and integrating the classification error over the entire 2D space) is just 1.4%. Thus, 98.6% of all points submitted to the classifier will be classified correctly despite the poor parameter estimation.

(rather than via *K-Means* clustering) from amongst the filter response vectors. In this case, the classification result for VZ MR8 drops by only 5% and is still well above 90%. A similar phenomenon was observed by [19] when *Mean-Shift* clustering was used to approximate the filter response PDF. Thus accurate parameter estimation does not seem to be essential for accurate texture classification and the loss due to inaccurate parameter estimation in high dimensions might be less than the loss associated with projecting into a lower dimensional subspace even though clustering may be improved.

6.2.2) Feature extraction: The main argument from feature extraction is that many features at multiple orientations and scales must be detected accurately for successful classification. Furthermore, studies of early vision mechanisms and pre-attentive texture discrimination have suggested that the detected features should look like edges, bars, spots and rings. These have most commonly come to be implemented using Gabor or Gaussian filters and their derivatives. However, results from the previous sections have shown that a multi-scale, multi-orientation large support filter bank is not necessary. Small image patches can also lead to successful classification. Furthermore, while an optimally designed bank might be maximising some measure of separability in filter space, it is hard to argue that “off the shelf” filters such as MR8, LM or S (whether biologically motivated or not) are the best for any given classification task. In fact, as has been demonstrated, a classifier which *learns* from all the input data present in an image patch should do better than one which depends on these pre-defined features bases.

It can also be argued that patch based features might not perform well in the presence of large, non-linear illumination changes. Edge based features, computed by thresholding filter responses, might be more stable in this case. However, the same effect can be achieved by putting a suitable prior over patches while learning the texton dictionary. Thresholding to find edges would then correspond to the vector quantisation step in our algorithm. Note that the CUReT database already contains images taken under significant illumination variation (see examples in Figure 1 as well as images of aluminium foil and leaves in the database). Nevertheless, it was noticed that the patch based classifiers gave better results than filter banks even when only a small number of training images was used. On a related note, patch based methods are also beginning to provide viable alternatives to filter banks for texture edge detection and segmentation tasks [56].

6.2.3) Noise reduction and invariance: Most filters have the desirable property that, because of their large smoothing kernels (such as Gaussians with large standard deviation), they are fairly robust to noise. This property is not shared by image patches. However, pre-processing the data can solve this problem. For example, the classifiers developed in this paper rely on vector quantisation of the patches into textons to help cope with noise. This can actually provide a superior alternative to filtering, because even though filters reduce noise, they also smooth the high frequency information present in the signal. Yet, as has been demonstrated in the 3×3 patch case, this information can be beneficial for classification. Therefore, if image patches can be denoised by pre-processing or quantisation without the loss of high frequency information then they should provide a superior representation for classification as compared to filter banks.

Virtually the same methods can be used to build invariance into the patch representation as are used for filters – without losing information by projecting onto lower dimensions. For example, patches are pre-processed and made to have zero mean and unit standard deviation to achieve invariance to affine transformations in the illuminant's intensity. Similarly, as discussed in Section 5.2, to achieve rotational invariance, the dominant orientation can be determined and used to orient the patch. This does have the drawback of being potentially unstable if the dominant direction cannot be determined accurately. For instance, corners have two dominant orientations and, in the presence of noise, can be transformed incorrectly upon reduction to the canonical frame. One solution to the problem could be to discard such ambiguous patches altogether. Another would be to take appropriately weighted linear combinations of all transformations. At the other extreme, one can even include many transformed copies of the patch (for instance, all rotated versions) in the training set to overcome this problem.

It should be noted that the arguments presented in this section do not imply that any arbitrary patch based classifier is better than every filter bank based one. We gave a constructive example of how local patches can make classification mistakes which can be avoided by larger scale filter banks. Furthermore, Figure 15 also illustrates how a filter bank, designed using prior knowledge, can perform better if there is limited training data. Instead, our arguments focus on two points. First, small local patch based classifiers can give surprisingly good results in many real world situations. Second, given equivalent prior knowledge, patches should do as well, if not better, than filter banks with *equivalent support*.

7. SYNTHESIS AND DENOISING

In this section, we investigate how accurately distributions in high dimensional spaces can be learnt given limited training data. The concern is that the high dimensional, patch based representation is incapable of capturing a texture's statistics as compared to lower dimensional filter responses. Most of the arguments in subsections 5.3 and 6.2 revolve around this central issue. While demonstrating good classification results is one way of addressing the concern, another way is to synthesise or denoise textures by sampling from the learnt PDF. If the reconstruction is adequate, then that provides additional evidence that our PDF representation is sufficiently accurate. Therefore, in this section, we also demonstrate that our MRF representation can be used to synthesise and denoise textures.

7.1. Texture Synthesis

Our texture synthesis algorithm is very similar to [15] but for the fact that we explicitly learn and sample from the texture's PDF. Given an input texture block to be synthesised, the first step is to learn its MRF statistics using the matrix representation of the PDF of image patches. The parameters that can be varied are N, the size of the neighbourhood, and K the number of textons used to represent the neighbourhood distribution. The central pixel PDF is stored in 256 bins in this case. Next, to synthesise the texture, the input block is initially tiled to the required dimensions. A new image is synthesised from this tiled image by taking every pixel, determining its neighbourhood (i.e. closest texton) and setting the pixel to a value randomly sampled from the learnt MRF distribution. This iteration is repeated until a desired synthesis is obtained. Results are shown in Figure 17. As can be seen, the synthesised textures are very similar to the originals, thereby indicating that the MRF representation can form an adequate representation of the texture's statistics. Note that no higher order statistics or image regularity information [34], [36] has been used and this can only improve results. Furthermore, once the

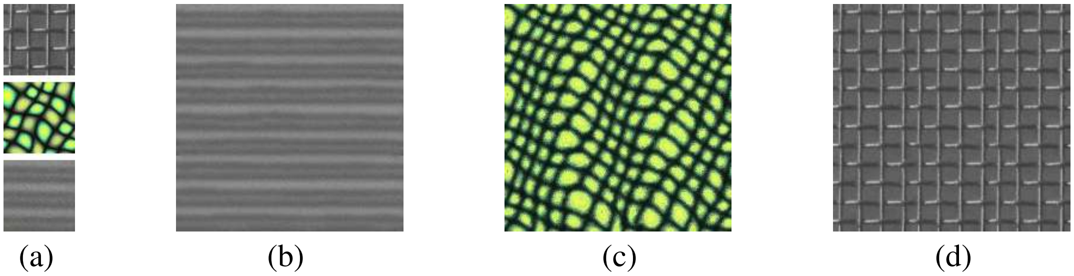

Fig. 17. Synthesis Results: (a) Input texture blocks, (b) Ribbed Paper (CUReT) synthesised using a 7×7 neighbourhood and 100 textons (c) Efros and Leung [15] - 15×15, 800 textons and (d) D6 (Brodatz) - 11×11, 300 textons.

representation has been learnt explicitly, the synthesised image can be generated quickly as exhaustive image search [15] is no longer required. On the other hand, a disadvantage of our method is that the neighbourhood is fixed while it can be adapted in [15].

7.2. Texture Denoising

Our denoising algorithm is inspired by [7]. In the first stage, the MRF representation of the noisy images is learnt exactly as had been done for synthesis. This involves clustering all $N \times N$ patches of the noisy image into K textons and then learning the central pixel PDF given each of the K textons. Denoising is carried out by labelling each patch in the noisy image by its closest texton and then replacing the central pixel in the patch by the median of the corresponding central pixel distribution (other statistics, such as the mean or the mode, can also be used if found to be more appropriate for a given noise model). As such, the algorithm is identical to our synthesis algorithm except that the pixels in the denoised image are generated by choosing the median of the appropriate central pixel PDF rather than sampling from it.

Figure 18 shows some typical results using $N = 7$ and $K = 1000$. The central pixel PDF was stored using 256 bins. No attempt was made at optimising these parameters. In contrast

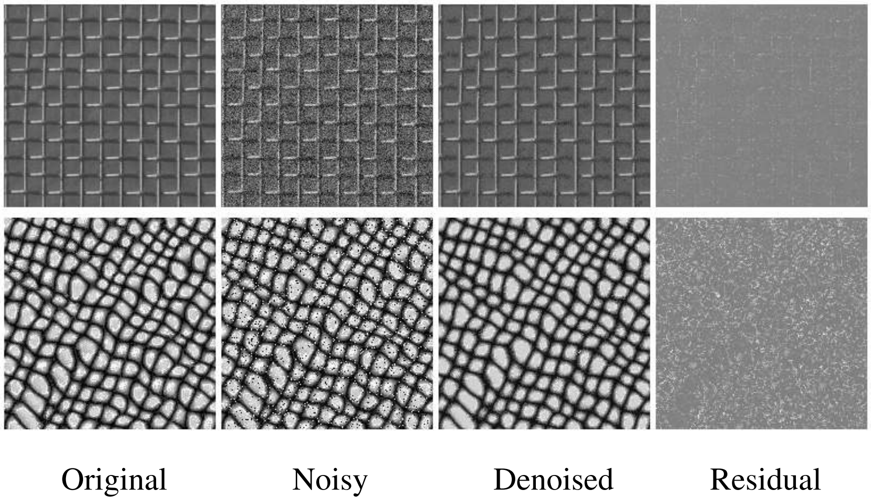

Fig. 18. Texture denoising using the MRF representation: In the top row, Gaussian white noise with $\sigma = 0.01$ was artificially added to the image. In the bottom row, salt and pepper noise with density 0.075 was added. Each image was denoised by first learning the MRF representation from the noisy image (using 7×7 patches and 1000 textons) and then replacing the value of the central pixel in every patch by the median of its distribution.

to [7], no Gaussian smoothing of the neighbourhood was performed. Furthermore, statistics were computed over the entire image rather than over a restricted subregion around the central pixel being denoised.

Again, the good results indicate that the high dimensional image patch PDF has been learnt accurately enough and that one does not have to reduce dimensionality using a filter bank in order to capture a texture's statistics.

8. Conclusions

We have described a classification method based on representing textures as a set of exemplar patches. This representation has been shown to be superior to one based on filters banks.

Filter banks have a number of disadvantages compared to smaller image patches: first, the large support they require means that far fewer samples of a texture can be learnt from training images (there are many more 3×3 neighbourhoods than 50×50 in an 100×100 image). Second, the large support is also detrimental in texture segmentation, where boundaries are localised less precisely due to filter support straddling region boundaries; A third disadvantage is that the blurring (e.g. Gaussian smoothing) in many filters means that fine local detail can be lost.

The disadvantage of the patch representation is the quadratic increase in the dimension of the feature space with the size of the neighbourhood. This problem may be tackled by using a multi-scale representation. For instance, an image pyramid could be constructed and patches taken from several layers of the pyramid if necessary. An alternative would be to use large neighbourhoods but store the pixel information away from the centre at a coarser resolution. A scheme such as Zalesny and Van Gool's [58] could also be implemented to determine which long range interactions were important and use only those cliques.

Before concluding, it is worth while to reflect on how the image patch algorithms and their results relate to what others have observed in the field. In particular, [16], [32], [40] have all noted that in their segmentation and classification tasks, filters with small support have outperformed the same filters at larger scales. In addition, [56] use small 5×5 patches to detect texture edges. Thus, there appears to be emerging evidence that small support is not necessarily detrimental to performance.

It is also worth noting that the "new" image patch algorithms, such as the synthesis method of Efros and Leung and the Joint classifier developed in this paper, have actually been around for quite a long time. For instance, Efros and Leung discovered a strong resemblance between their algorithm and that of [17]. Furthermore, both the Joint classifier and Efros and Leung's algorithm are near identical in spirit to the work of Popat and Picard [39]. The relationship between the Joint classifier and Popat and Picard's algorithm is particularly close as both use clustering to learn a distribution over image patches which then forms a model for novel texture classification. Apart from the choice of neighbourhoods, the only minor differences between the two methods are in the representation of the PDF and the distance measure used during classification. Popat and Picard use a Gaussian mixture model with diagonal covariances to represent their PDF while the texton representation used in this paper can be thought of as fitting a spherical Gaussian mixture model via *K-Means*. During classification, Popat and Picard use a *naïve* Bayesian method which, for the Joint classifier, would equate to using nearest neighbour matching with KL divergence instead of the χ^2 statistic as the distance measure [53].

Certain similarities also exist between the Joint classifier and the MRF model of Cross and Jain [10]. In particular, Cross and Jain were the first to recommend that χ^2 over the distribution of central pixels and their neighbours could be used to determine the best fit between a sample texture and a model. Had they actually used this for classification rather than just model validation of synthesised textures, the two algorithms would have been very similar apart from the functional form of the PDFs learnt (Cross and Jain treat the conditional PDF of the central pixel given the neighbourhood as a unimodal binomial distribution).

Thus, alternative approaches to filter banks have been around for quite some time. Perhaps the reason that they didn't become popular then was due to the computational costs required to achieve good results. For instance, the synthesis results of [39] are of a poor quality which is perhaps why their theory didn't attract the attention it deserved. However, with computational power being readily accessible today, MRF and image patch methods are outperforming filter bank based methods.

Acknowledgements

We are grateful to Alexey Zalesny, David Forsyth and Andrew Fitzgibbon for many discussions and valuable feedback. We would like to thank Varun Gulshan for helping with the CUReT and UIUC experiments. We would also like to thank Alan Yuille for making available the San Francisco database, and Antonio Criminisi and Silvio Savarese for the Microsoft Textile database.

The investigations reported in this contribution have been supported by a University of Oxford Graduate Scholarship in Engineering at Jesus College, an ORS award and the European Union (FP5-project ‘CogViSys’, IST-2000-29404).

References

[1] http://www.robots.ox.ac.uk/ vgg/research/texclass/data/curetcol.zip.

[2] http://www.robots.ox.ac.uk/˜vgg/research/texclass/filters.html.

[3] J. R. Bergen and E. H. Adelson. Early vision and texture perception. *Nature*, 333:363–364, May 1988.

[4] J. Bi, K. Bennett, M. Embrechts, M. Breneman, and M. Song. Dimensionality reduction via sparse support vector machines. *Journal of Machine Learning Research*, 3:1229–1243, 2003.

[5] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[6] R. E. Broadhurst. Statistical estimation of histogram variation for texture classification. In *Proceedings of the Fourth International Workshop on Texture Analysis and Synthesis*, pages 25–30, Beijing, China, October 2005.

[7] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 60–65, San Diego, California, June 2005.

[8] C. J. C. Burges. Geometric methods for feature extraction and dimensionality reduction. In O. Maimon and L. Rokach, editors, *Data Mining and Knowledge Discovery Handbook*, pages 59–92. Springer, 2005.

[9] B. Caputo, E. Hayman, and P. Mallikarjuna. Class-specific material categorisation. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1597–1604, Beijing, China, October 2005.

[10] G. K. Cross and A. K. Jain. Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(1):25–39, November 1983.

[11] O. G. Cula and K. J. Dana. 3D texture recognition using bidirectional feature histograms. *International Journal of Computer Vision*, 59(1):33–60, August 2004.

[12] K. J. Dana, B. van Ginneken, S. K. Nayar, and J. J. Koenderink. Reflectance and texture of real world surfaces. *ACM Transactions on Graphics*, 18(1):1–34, January 1999.

[13] P. Domingos and M. J. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130, November 1997.

[14] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons, second edition, 2001.

[15] A. Efros and T. Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1039–1046, Corfu, Greece, September 1999.

[16] C. Fowlkes, D. Martin, and J. Malik. Learning affinity functions for image segmentation: Combining patch-based and gradient-based approaches. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 54–61, Madison, Wisconsin, June 2003.

[17] D. D. Garber. *Computational Models for Texture Analysis and Texture Synthesis*. PhD thesis, University of Southern California, 1981.

[18] S. Geman and D Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984.

[19] B. Georgescu, I. Shimshoni, and P. Meer. Mean shift based clustering in high dimensions: A texture classification example. In *Proceedings of the International Conference on Computer Vision*, volume 1, pages 456–463, Nice, France, October 2003.

[20] H. Greenspan, S. Belongie, P. Perona, and R. Goodman. Rotation invariant texture recognition using a steerable pyramid. In *Proceedings of the International Conference on Pattern Recognition*, volume 2, pages 162–167, October 1994.

[21] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[22] G. M. Haley and B. S. Manjunath. Rotation-invariant texture classification using modified gabor filters. In *Proceedings of the IEEE International Conference on Image Processing*, volume 1, pages 262–265, Washington, DC, October 1995.

[23] E. Hayman, B. Caputo, M. Fritz, and J-O. Eklundh. On the significance of real-world conditions for material classification. In *Proceedings of the European Conference on Computer Vision*, volume 4, pages 253–266, Prague, Czech Republic, May 2004.

[24] J. Hays, M. Leordeanu, A. Efros, and Y. Liu. Discovering texture regularity as a higher-order correspondence problem. In *Proceedings of the European Conference on Computer Vision*, volume 2, pages 522–535, 2006.

[25] A. K. Jain and K. Karu. Learning texture discrimination masks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):195–205, February 1996.

[26] B. Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290:91–97, 1981.

[27] B. Julesz, E. N. Gilbert, L. A. Shepp, and H. L. Frisch. Inability of humans to discriminate between visual textures that agree in second-order statistics – revisited. *Perception*, 2(4):391–405, 1973.

[28] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[29] S. Konishi and A. L. Yuille. Statistical cues for domain specific image segmentation with performance analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 125–132, Hilton Head, South Carolina, June 2000.

[30] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1265–1278, August 2005.

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

[32] E. Levina. *Statistical Issues in Texture Analysis*. PhD thesis, University of California at Berkeley, 2002.

[33] S. Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer-Verlag, 2001.

[34] W. Lin, J. Hays, C. Wu, V. Kwatra, and Y. Liu. Quantitative evaluation of near regular texture synthesis algorithms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 427–434, 2006.

[35] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):77–116, November 1998.

[36] Y. Liu, W. Lin, and J. Hays. Near regular texture analysis and manipulation. *ACM Transactions on Graphics*, 23(3):368–376, 2004.

[37] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1):7–27, June 2001.

[38] J. Malik and P. Perona. Preattentive texture discrimination with early vision mechanism. *Journal of the Optical Society of America*, 7(5):923–932, May 1990.

[39] K. Popat and R. W. Picard. Novel cluster-based probability model for texture synthesis, classification, and compression. In *Proceedings of the SPIE Conference on Visual Communication and Image Processing*, pages 756–768, Boston, Massachusetts, November 1993.

[40] T. Randen and J. H. Husoy. Filtering for texture classification: A comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):291–310, April 1999.

[41] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025, November 1999.

[42] S. Savarese and A. Criminsi. Classification of folded textiles, August 2004. http://research.microsoft.com/vision/cambridge/recognition/MSRC_MaterialsImageDatabase.zip.

[43] C. Schmid. Constructing models for content-based image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 39–45, Kauai, Hawaii, December 2001.

[44] C. Schmid. Weakly supervised learning of visual models and its application to content-based retrieval. *International Journal of Computer Vision*, 56(1):7–16, 2004.

[45] B. Scholkopf and A. Smola. *Learning with Kernels*. MIT Press, 2002.

[46] P. Simard, Y. LeCun, J. Denker, and B. Victorri. Transformation invariance in pattern recognition – tangent distance and tangent propagation. *International Journal of Imaging System and Technology*, 11(2):181–194, 2001.

[47] J. R. Smith and S. F. Chang. Transform features for texture classification and discrimination in large image databases. In *Proceedings of the IEEE International Conference on Image Processing*, volume 3, pages 407–411, Austin, Texas, November 1994.

[48] K. Tieu and P. Viola. Boosting image retrieval. *International Journal of Computer Vision*, 56(1-2):17–36, 2004.

[49] M. Unser. Local linear transforms for texture measurements. *Signal Processing*, 11(1):61–79, 1986.

[50] M. Varma. *Statistical Approaches To Texture Classification*. PhD thesis, University of Oxford, October 2004.

[51] M. Varma and R. Garg. Locally invariant fractal features for statistical texture classification. In *Proceedings of the International Conference on Computer Vision*, 2007.

[52] M. Varma and A. Zisserman. Texture classification: Are filter banks necessary? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 691–698, Madison, Wisconsin, June 2003.

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

[54] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *International Journal of Computer Vision*, 2005.

[55] C. Wallraven, B. Caputo, and A. Graf. Recognition with local features: the kernel recipe. In *Proceedings of the International Conference on Computer Vision*, volume 1, pages 257–264, Nice, France, October 2003.

[56] L. Wolf, X. Huang, I. Martin, and D. Metaxas. Patch based texture edges and segmentation. In *Proceedings of the European Conference on Computer Vision*, volume 2, pages 481–493, Graz, Austria, May 2006.

[57] Y. Xu, H. Ji, and C. Fermuller. A projective invariant for textures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1932–1939, New York, New York, June 2006.

[58] A. Zalesny and L. Van Gool. A compact model for viewpoint dependent texture synthesis. In *Proceedings of the European Workshop on 3D Structure from Multiple Images of Large-Scale Environments*, pages 124–143, Dublin, Ireland, 2000.