A Statistical Approach to Texture Classification from Single Images

Manik Varma and Andrew Zisserman
Robotics Research Group
Dept. of Engineering Science
University of Oxford
Oxford, OX1 3PJ, UK
(manik,az)@robots.ox.ac.uk

Abstract. We investigate texture classification from single images obtained under unknown viewpoint and illumination. A statistical approach is developed where textures are modelled by the joint probability distribution of filter responses. This distribution is represented by the frequency histogram of filter response cluster centres (textons). Recognition proceeds from single, uncalibrated images and the novelty here is that rotationally invariant filters are used and the filter response space is low dimensional.

Classification performance is compared with the filter banks and methods of Leung and Malik [IJCV 2001], Schmid [CVPR 2001] and Cula and Dana [IJCV 2004] and it is demonstrated that superior performance is achieved here. Classification results are presented for all 61 materials in the Columbia-Utrecht texture database.

We also discuss the effects of various parameters on our classification algorithm – such as the choice of filter bank and rotational invariance, the size of the texton dictionary as well as the number of training images used. Finally, we present a method of reliably measuring relative orientation co-occurrence statistics in a rotationally invariant manner, and discuss whether incorporating such information can enhance the classifier's performance.

Keywords: material classification, 3D textures, textons, filter banks, rotation invariance

1. Introduction

In this paper, we investigate the problem of classifying materials from their imaged appearance, without imposing any constraints on, or requiring any *a priori* knowledge of, the viewing or illumination conditions under which these images were obtained. Classifying textures from single images under such general conditions is a very demanding task.

A texture image is primarily a function of the following variables: the texture surface, its albedo, the illumination, the camera and its viewing position. Even if we were to keep the first two parameters fixed, i.e. photograph exactly the same patch of texture every time, minor changes in the other parameters can lead to dramatic changes in


© 2004 *Kluwer Academic Publishers. Printed in the Netherlands.*

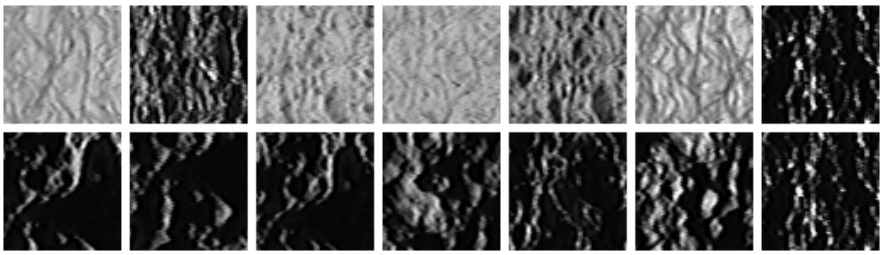


Figure 1. The change in imaged appearance of the same texture (Plaster B, texture # 30 from the Columbia-Utrecht database) with variation in imaging conditions. Top row: constant viewing angle and varying illumination. Bottom row: constant illumination and varying viewing angle. There is a considerable difference in the appearance across images.

the resultant image (see figure 1). This causes a large variability in the imaged appearance of a texture and dealing with it successfully is one of the main tasks of any classification algorithm. Another factor which comes into play is that, quite often, two textures when photographed under very different imaging conditions can appear to be quite similar, as is illustrated by figure 2. It is a combination of both these factors which makes the texture classification problem so hard.

A statistical learning approach to the problem is developed and investigated in this paper. Textures are modelled by the joint distribution of filter responses. This distribution is represented by texton (cluster centre) frequencies, and textons and texture models are learnt from training images. Classification of a novel image proceeds by mapping the image to a texton distribution and comparing this distribution to the learnt models. As such, this procedure is quite standard (Leung and Malik, 2001), but the originality comes in at two points: first, texton

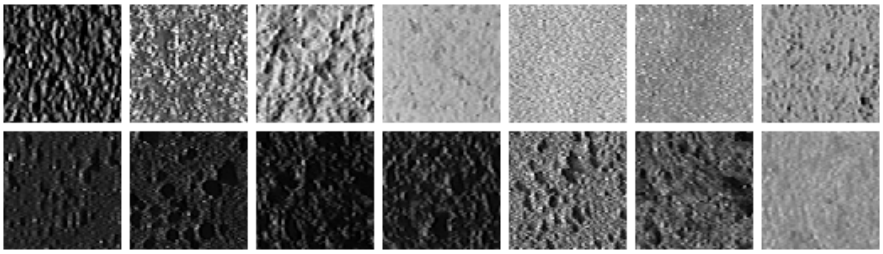


Figure 2. Small inter class variations between textures can make the problem harder still. In the top row, the first and the fourth image are of the same texture while all the other images, even though they look similar, belong to different classes. Similarly, in the bottom row, the images appear similar and yet there are three different texture classes present.

clustering is in a very low dimensional space and is also rotationally invariant. The second innovation is to classify textures from single images while representing each texture class by a small set of models.

Our approach is most closely related to those of Leung and Malik (Leung and Malik, 2001), Schmid (Schmid, 2001) and Cula and Dana (Cula and Dana, 2004). Leung and Malik's method is not rotationally invariant and requires as input a set of registered images acquired under a (implicitly) known set of imaging conditions. Schmid's approach is rotationally invariant but the invariance is achieved in a different manner to ours, and texton clustering is in a higher dimensional space. Cula and Dana classify from single images, but the method is not rotationally invariant and their algorithm for model selection differs from the one developed in this paper. These points are discussed in more detail subsequently.

The paper is organised as follows: in section 2, the basic classification algorithm is developed within a rotationally invariant framework. The clustering, learning and classification steps of the algorithm are described, and the performance of four filter sets is compared. The sets include those used by Schmid (Schmid, 2001), Leung and Malik (Leung and Malik, 2001), and two rotationally invariant sets based on maximal filter responses. In section 3, methods are developed which minimise the number of models used to characterise the various texture classes. Section 4 then deals with various modifications and generalisations of the basic algorithm. In particular, the effect of the choice of texton dictionary and training images upon the classifier is investigated. Finally, the issue of whether information is lost by using only the first order statistics of rotationally invariant filter responses is discussed. A method for reliably measuring the relative orientation co-occurrence of textons is presented in order to incorporate second order statistics into the classification scheme.

All experiments are carried out on the Columbia-Utrecht (CUReT) database (Dana et al., 1999), the same database used by (Cula and Dana, 2004; Leung and Malik, 2001). It is demonstrated that the classifier developed here achieves performance superior to that of (Cula and Dana, 2004) and (Leung and Malik, 2001), while requiring only a single image as input and with no information (implicit or explicit) about the illumination and viewing conditions. The CUReT database contains 61 textures, and each texture has 205 images obtained under different viewing and illumination conditions. The variety of textures in this database is shown in figure 3. Results are reported for all 61 textures. A preliminary version of these results appeared in (Varma and Zisserman, 2002).

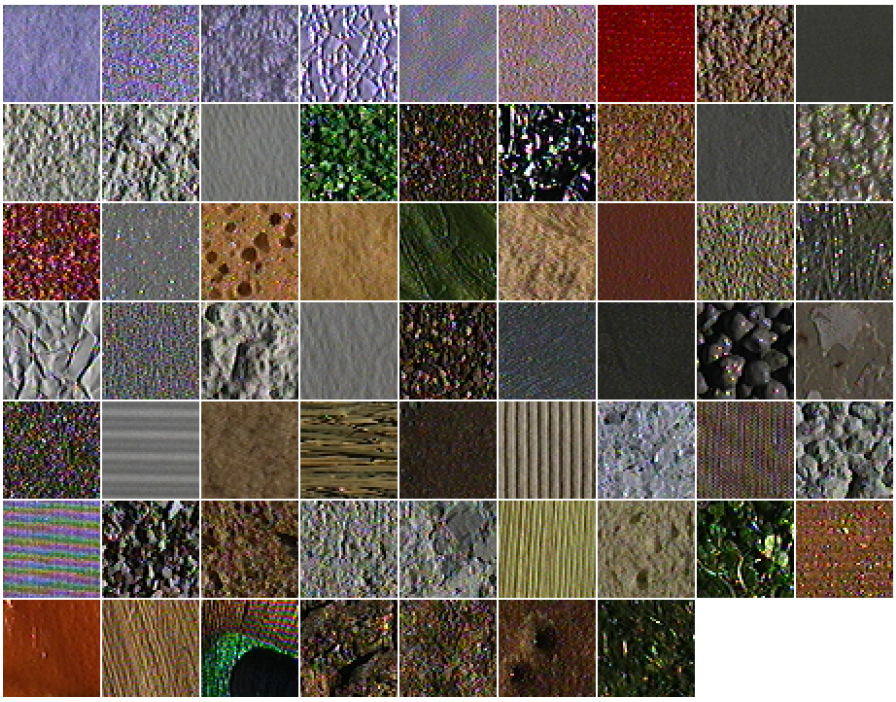


Figure 3. Textures from the Columbia-Utrecht database. All images are converted to monochrome in this work, so colour is not used in discriminating different textures.

1.1. Background

Most of the early work on material classification tended to view texture as albedo variation on a flat surface – thereby ignoring all surface normal effects which play a major role when imaging conditions vary. Recently, however, focus has been placed on these surface normal, or *3D*, effects. Chantler *et al.* (Chantler et al., 2002a; Chantler et al., 2000; Chantler et al., 2002b) and Penirschke *et al.* (Penirschke et al., 2002) have studied the effect of change in illumination on textures and have developed photometric stereo based classification algorithms.

Dana *et al.* (Dana et al., 1999), realising the need for a large texture database which captured the variation of imaged appearances with change in viewpoint and illumination, created the Columbia-Utrecht (CUReT) database. Dana and Nayar (Dana and Nayar, 1998; Dana and Nayar, 1999) developed parametric models based on surface roughness and correlation lengths which were tested on sample textures from the CUReT database. However, no significant classification results were presented.

Leung and Malik (Leung and Malik, 2001) were amongst the first to seriously tackle the problem of classifying textures under varying viewpoint and illumination. In particular, they made an important innovation by giving an operational definition of a texton. They defined a 2D texton as a cluster centre in filter response space. This not only enabled textons to be generated automatically from an image, but also opened up the possibility of a *universal* set of textons for all images. To compensate for 3D effects, they proposed 3D textons which were cluster centres of filter responses over a stack of images with representative viewpoints and lighting. In the learning stage of their classification algorithm, 20 images of each texture were geometrically registered and mapped to a 48 dimensional filter response space. The registration was necessary because the clustering that defined the texton was in the stacked $20 \times 48 = 960$ dimensional space (i.e. the textons were 960-vectors), and it was important that each filter be applied at the same texture surface point as camera pose and illumination varied. In the classification stage, 20 novel images of the same texture were presented. However, these images also had to be registered and more significantly had to have the same order as the original 20 (i.e. they had to be taken from images with similar viewpoint and illumination to the original). In essence, the viewpoint and lighting were being supplied implicitly by this ordering. Leung and Malik also developed an MCMC algorithm for classifying a single image under *known* imaging conditions. However, the classification accuracy of this method was not as good as that achieved by the multiple image method.

Cula and Dana (Cula and Dana, 2004) presented an algorithm based on Leung and Malik's framework but capable of classifying single images without requiring any *a priori* information. Using much the same filter bank as Leung and Malik, they showed how to achieve results comparable to (Leung and Malik, 2001) but using 2D textons generated from single images instead of registered image stacks. We compare the performance of our algorithm with theirs in section 3.

Suen and Healy (Suen and Healey, 2000) used correlation functions across multiple colour bands to determine basis textures for each texture class. They assumed that, for every texture image picked from a given class, the correlation function for that image could be represented as a linear combination of the basis texture correlation functions of that class. A nearest neighbour classifier employing the sum of squared differences metric was used. The number of basis images for a particular texture class also provided information about the *dimensionality* of that class. The main drawback of their algorithm was its heavy reliance on colour rather than purely on texture. While colour provides a very strong cue for discrimination, it can also be misleading due to the colour

constancy problem (Funt et al., 1998). The classifier developed in this paper does not use colour information at all but rather normalises the images and filter responses so as to achieve partial invariance to changes in illuminant intensity.

2. The Basic Algorithm

Weak classification algorithms based on the statistical distribution of filter responses have been particularly successful of late (Cula and Dana, 2004; Konishi and Yuille, 2000; Leung and Malik, 2001; Schmid, 2001). Our classification algorithm too is one such and, as is customary amongst weak classifiers, is divided into a learning stage and a classification stage. In the learning stage, training images are convolved with a filter bank to generate filter responses (see figure 4). Exemplar filter responses are chosen as *textons* (via *K-Means* clustering (Duda et al., 2001)) and are used to label each filter response, and thereby every pixel, in the training images. The histogram of texton frequencies is then used to form *models* corresponding to the training images (see figure 5). In the classification stage, the same procedure is followed to build the histogram corresponding to the novel image. This histogram is

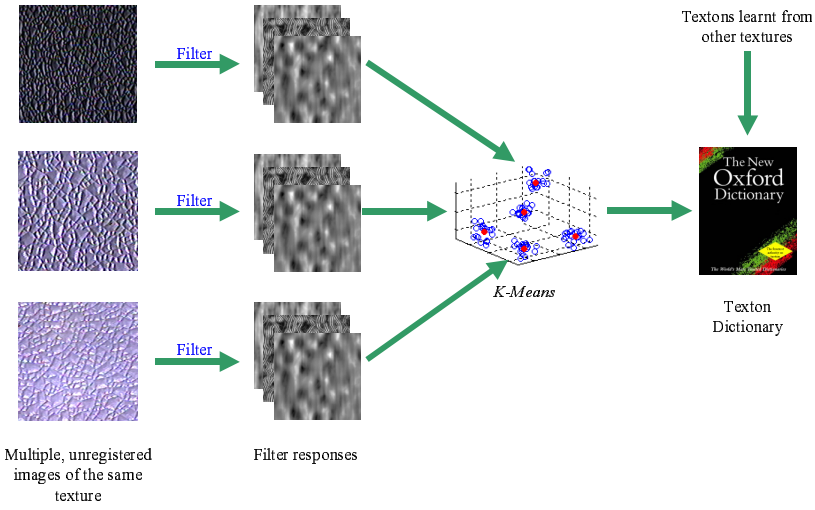


Figure 4. **Learning stage I: Generating the texton dictionary**. Multiple, unregistered images from the training set of a particular texture class are convolved with a filter bank. The resultant filter responses are aggregated and clustered into textons using the *K-Means* algorithm. Textons from different texture classes are combined to form the texton dictionary.

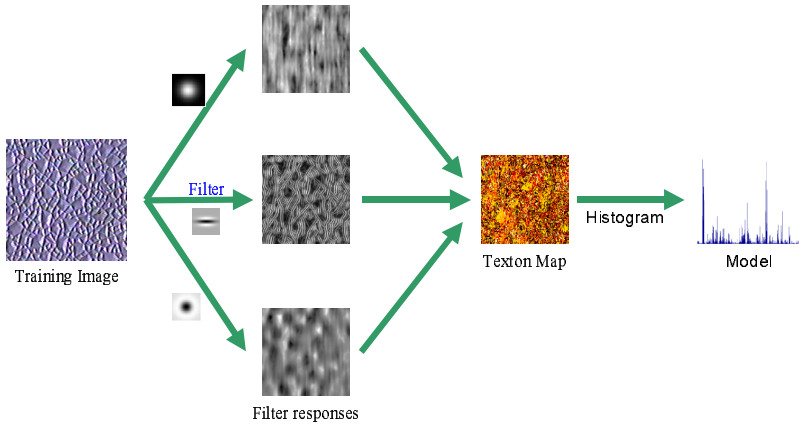


Figure 5. **Learning stage II: Model generation**. Given a training image, its corresponding model is generated by first convolving it with a filter bank and then labelling each filter response with the texton which lies closest to it in filter response space. The histogram of textons, i.e. the frequency with which each texton occurs in the labelling, forms the model corresponding to the training image.

then compared with the models learnt during training and is classified on the basis of the comparison (see figure 6). A nearest neighbour classifier is used and the χ^2 statistic employed to measure distances. The histograms should be normalised to sum to unity, but this is not required in our case as all training and testing images have the same number of pixels.

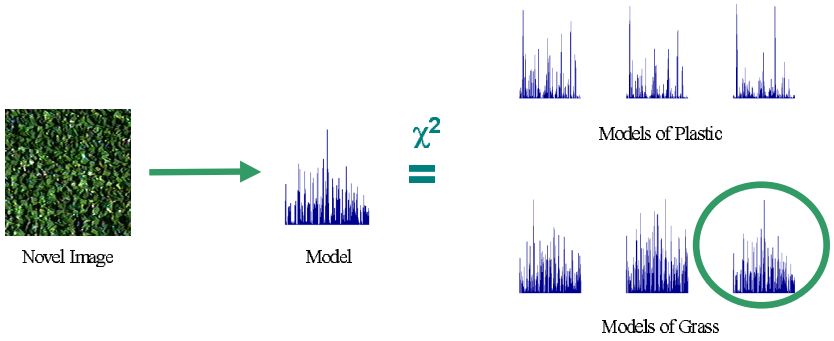


Figure 6. **Classification stage**. A novel image is classified by forming its histogram and then using a nearest neighbour classifier to pick the closest model to it (in the χ^2 sense). The novel image is declared as belonging to the texture class of the closest model.

In the following subsections, we describe the filters and algorithmic steps in more detail. Classification results are presented on the CUReT database, and compared with those of Leung and Malik (Leung and Malik, 2001) and Cula and Dana (Cula and Dana, 2004).

2.1. Rotationally invariant filters

In this subsection, we introduce the rotationally invariant filter sets that are used in the classification algorithm. We also describe two other filter sets that will be used in classification comparisons in subsection 2.4. The aspects of interest are the dimension of the filter space, and whether the filter set is rotationally invariant or not.

The four filter sets that will be compared are: those of Leung and Malik (Leung and Malik, 2001) which are not rotationally invariant; those of Schmid (Schmid, 2001) which are; and two reduced sets of filters based on using the maximum response (which are again rotationally invariant). Filter sets will be assessed by their classification performance using textons clustered in their response spaces.

2.1.1. *The Leung-Malik (LM) set*

The LM set consists of 48 filters, partitioned as follows: first and second derivatives of Gaussians at 6 orientations and 3 scales making a total of 36; 8 Laplacian of Gaussian filters; and 4 Gaussians. The scale of the filters range between $\sigma = 1$ and $\sigma = 10$ pixels. They are shown in figure 7.

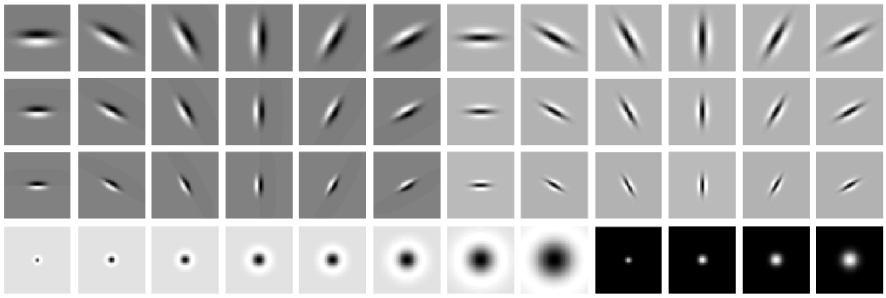


Figure 7. The LM filter bank has a mix of edge, bar and spot filters at multiple scales and orientations. It has a total of 48 filters - 2 Gaussian derivative filters at 6 orientations and 3 scales, 8 Laplacian of Gaussian filters and 4 Gaussian filters.

2.1.2. *The Schmid (S) set*

The S set consists of 13 rotationally invariant filters of the form

$$F(r, \sigma, \tau) = F_0(\sigma, \tau) + cos\left(\frac{\pi\tau r}{\sigma}\right) e^{-\frac{r^2}{2\sigma^2}}$$

where $F_0(\sigma, \tau)$ is added to obtain a zero DC component with the (σ, τ) pair taking values (2,1), (4,1), (4,2), (6,1), (6,2), (6,3), (8,1), (8,2), (8,3), (10,1), (10,2), (10,3) and (10,4). The filters are shown in figure 8. As can be seen all the filters have rotational symmetry.

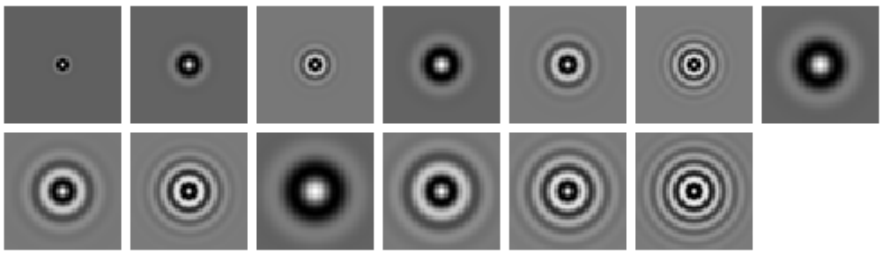


Figure 8. The S filter bank is rotationally invariant and has 13 isotropic, "Gabor-like" filters.

2.1.3. *The Maximum Response (MR) sets*
The MR8 filter bank consists of 38 filters but only 8 filter responses. The filter bank contains filters at multiple orientations but their outputs are "collapsed" by recording only the maximum filter response across all orientations. This achieves rotation invariance. The filter bank is shown in figure 9 and consists of a Gaussian and a Laplacian of Gaussian both with $\sigma = 10$ pixels (these filters have rotational symmetry), an edge filter at 3 scales (σ_x,σ_y)={(1,3), (2,6), (4,12)} and a bar filter at the same 3 scales. The latter two filters are oriented and, as in LM, occur at 6 orientations at each scale. Measuring only the maximum response across orientations reduces the number of responses from 38 (6 orientations at 3 scales for 2 oriented filters, plus 2 isotropic) to 8 (3 scales for 2 filters, plus 2 isotropic).

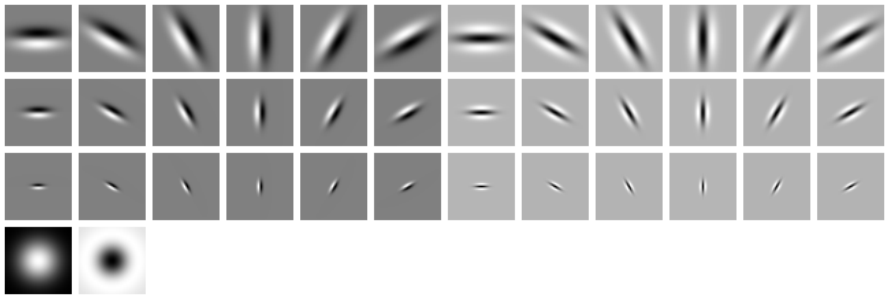


Figure 9. The MR8 filter bank consists of 2 anisotropic filters (an edge and a bar filter, at 6 orientations and 3 scales), and 2 rotationally symmetric ones (a Gaussian and a Laplacian of Gaussian). However only 8 filter responses are recorded by taking, at each scale, the maximal response of the anisotropic filters across all orientations.

The MR4 filter bank is a subset of the MR8 filter bank where the oriented edge and bar filters occur at a single fixed scale ($\sigma_x = 4$, $\sigma_y = 12$).

The motivation for introducing these MR filters sets is twofold. The first is to overcome the limitations of traditional rotationally invariant filters which do not respond strongly to oriented image patches and thus do not provide good features for anisotropic textures. However, since the MR sets contain both isotropic filters as well as anisotropic filters at multiple orientations they are expected to generate good features for all types of textures. Additionally, unlike traditional rotationally invariant filters, the MR sets are also able to record the angle of maximum response. This enables us to compute higher order co-occurrence statistics on orientation and such statistics may prove useful in discriminating textures which appear to be very similar. We return to this in subsection 4.2.

The second motivation arises out of a concern about the dimensionality of the filter response space. Quite apart from the extra processing and computational costs involved, the higher the dimensionality, the harder the clustering problem. In general, not only does the number of cluster centres needed to cover the space rise dramatically, so does the amount of training data required to reliably estimate each cluster centre. This is mitigated to some extent by the fact that texture features are sparse and can lie in lower dimensional subspaces. However, the presence of noise and the difficulty in finding and projecting onto these lower dimensional subspaces can counter these factors. Therefore, it is expected that the MR filter banks should generate more significant textons not only because of improved clustering in a lower dimensional space but also because rotated features are correctly mapped to the same texton.

2.2. Pre-processing

The following pre-processing steps are applied before going ahead with any learning or classification.

First, before convolving with any of the filter banks, a central 200×200 texture region is cropped and retained from every image and the extraneous background data discarded. All processing is done on these cropped regions and they are converted to grey scale and intensity normalised to have zero mean and unit standard deviation. This normalisation gives invariance to global (i.e. across the entire region) affine transformations in the illumination intensity.

Second, all 4 filter banks are L_1 normalised so that the responses of each filter lie roughly in the same range. In more detail, each filter F_i in

the filter bank is divided by $\|F_i\|_1$ so that the filter has unit L_1 norm. This helps vector quantization, when using Euclidean distances, as the scaling for each of the filter response axes becomes the same (Malik et al., 2001).

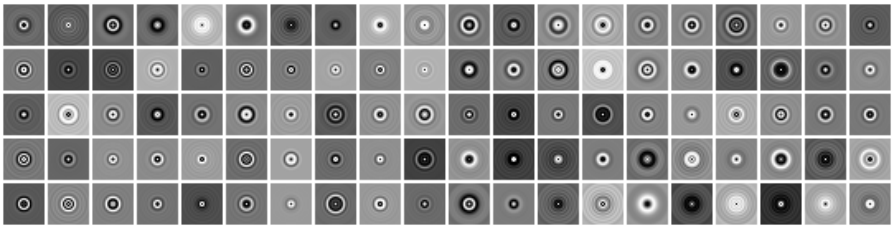
Third, following (Fowlkes et al., 2002; Malik et al., 2001) and motivated by Weber's law, the filter response at each pixel $\mathbf{x}$ is (contrast) normalised as

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

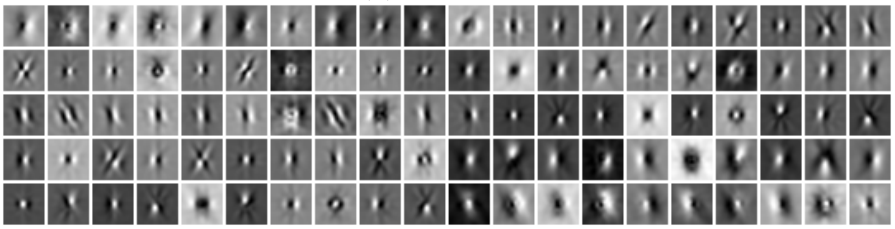
where $L(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|_2$ is the magnitude of the filter response vector at that pixel.

2.3. Textons by clustering

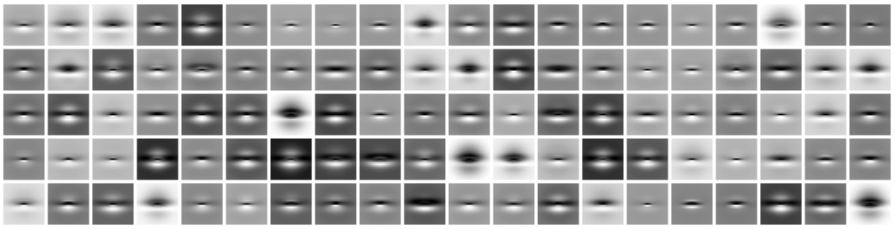
We now consider clustering the filter responses in order to generate a texton dictionary. This dictionary will subsequently be used to define texture models based on texton frequencies learnt from training images.



(a) S Textons



(b) LM Textons



(c) MR8 Textons

Figure 10. The first 100 textons recovered from 20 training textures using 13 images per texture: (a) S textons. (b) LM textons. (c) MR8 textons. Note that the LM textons are not rotationally symmetric.

For each filter set, we adopt the following procedure for computing a texton dictionary: A selection of 13 images is chosen randomly for each texture (these images sample the variations in illumination and viewpoint), the filter responses over all these images are aggregated, and 10 texton cluster centres computed using the standard *K-Means* algorithm (Duda et al., 2001). The learnt textons for each texture are then collected into a single dictionary. For example, if there are 5 texture classes then the dictionary will contain 50 textons. Examples of the textons for the S, LM and MR8 filter banks are shown in figure 10.

Our clustering task is considerably simpler than that of Leung and Malik, and Cula and Dana (who use essentially the same filter bank) as we are able to cluster in low, 4 and 8, dimensional spaces. This compares to 13 dimensional for S, and 48 dimensional for LM (we are not considering 3D textons at this point where the dimensionality is 960).

Concerning the rotation properties of the LM and MR textons, consider a texture and an (in plane) rotated version of the same texture. Corresponding features in the original and the rotated texture will map to the same point in MR filter space, but to different points in LM. It is therefore expected that more significant clusters will be obtained in the rotationally invariant case. Secondly, for the LM filter set, which is not rotationally invariant, it would be expected that its textons can not classify a rotated version of a texture unless the rotated version is included in the training set (both of these points are demonstrated in figure 11).

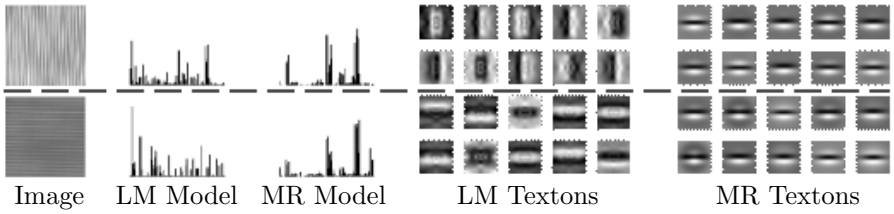


Figure 11. Classification of rotated textures. Two rotated images of Ribbed Paper have been taken from the CUReT database (texture numbers 38 and 38B) and their corresponding models generated using the LM and MR4 filter banks. Note that the MR models are very similar while the LM models are not. Therefore, in the case of MR, it is expected that by having one image present in the training set the other will be classified correctly. However, this will not hold true for LM as its models are quite dissimilar. Also note, that since the LM filter bank is not rotationally invariant, the textons that are generated by the two images are rotated copies of each other while, for MR, they are essentially the same.

This establishes that there is an advantage in being rotationally invariant as rotated versions of the same texture can be represented by

one histogram, while several are required for the LM textons. However, there is still the possibility that rotation invariance has the disadvantage that two different textures (which are not rotationally related) have the same histogram. We address this point next, where we compare classification rates over a variety of textures.

2.4. Classification Method and Comparison Results

In this subsection we perform three experiments to assess texture classification rates over 92 images for each of 20, 40 and 61 texture classes respectively. The first experiment, where we classify images from 20 textures, corresponds to the setup employed by Cula and Dana (Cula and Dana, 2004). The second experiment, where 40 textures are classified, is modelled on the setup of Leung and Malik (Leung and Malik, 2001). In the third experiment, we classify *all* 61 textures present in the Columbia-Utrecht database. The 92 images are selected as follows: for each texture in the database, there are 118 images where the viewing angle θ_v is less than 60 degrees. Out of these, only those 92 are chosen for which a sufficiently large region could be cropped across all texture classes.

Each experiment consists of three stages: texton dictionary generation; model generation, where texture models are learnt from training images; and, classification of novel images. The 92 images for each texture are partitioned into two, disjoint sets. Images in the first (training) set are used for dictionary and model generation, classification accuracy is only assessed on the 46 images for each texture in the second (test) set.

Each of the 46 training images per texture defines a model for that class as follows: the image is mapped (vector quantized) to a texton distribution (histogram). Thus, each texture class is represented by a set of 46 histograms. An image from the test set is classified by forming its histogram and then choosing the closest model histogram learnt from the training set. The distance function used to define closest is the χ^2 statistic (Press et al., 1992).

In all three experiments we follow both (Cula and Dana, 2004) and (Leung and Malik, 2001), and learn the texton dictionary from 20 textures (using the procedure outlined before in section 2.3). The particular textures used are specified in figure 7 of (Leung and Malik, 2001).

In the first experiment, 20 novel textures are chosen (see figure 19a in (Cula and Dana, 2004) for a list of the novel textures) and $20 \times 46 = 920$ novel images are classified in all. In the second experiment, the 40 textures specified in figure 7 of (Leung and Malik, 2001) are chosen and

a total of $40 \times 46 = 1840$ novel images classified. Finally, in the third experiment, all 61 textures in the Columbia-Utrecht database are classified using the same procedure. The results for all three experiments are presented in table I.

	# of texture classes		
filters	20	40	61
S	96.30%	95.27%	94.62%
LM	96.08%	93.75%	93.44%
MR4	94.13%	92.07%	90.73%
MR8	97.83%	96.41%	96.40%

Table I. Comparison of the classification rates for varying number of texture classes for each of the four filter sets. In all cases, a dictionary of 200 textons learnt from 20 textures is used and there are 46 models per texture class.

2.4.1. *Discussion*

Two points are notable in these results. First, the MR8 and S filters out perform the LM filters. This is a clear indicator that a rotationally invariant description is not a disadvantage (i.e. salient information for classification is not lost). Second, the fact that MR8 does better than S and LM is also evidence that it is detecting better features, for both isotropic and anisotropic textures, and that clustering in a lower dimensional space can be advantageous. The MR4 filter bank loses out because it only contains filters at a single scale and hence can't extract such rich features. What is also very encouraging with these results is that as the number of texture classes increases there is only a small decrease in the accuracy of the classifier.

3. Reducing the number of models

In this section, our objective is to reduce the number of training models required to characterise each texture class. In the previous section, the number of models was the same as the number of training images (and in effect (Leung and Malik, 2001) used 20 models/images for every texture). Here, we want to reduce the number of models to that appropriate for each class, independent of the number of training images.

One would expect that the number of different models that are needed to characterise a texture is a function of how much the texture changes in appearance with imaging conditions, i.e. it is a function

of the material properties of the texture. For example, if a texture is isotropic then the effect of varying the lighting azimuthal angle will be less pronounced than for one that is anisotropic. Thus, other parameters (such as relief profile) being equal, fewer models would be required for the isotropic texture (than the anisotropic) to cover the changes due to lighting variation. This is demonstrated in figure 12.

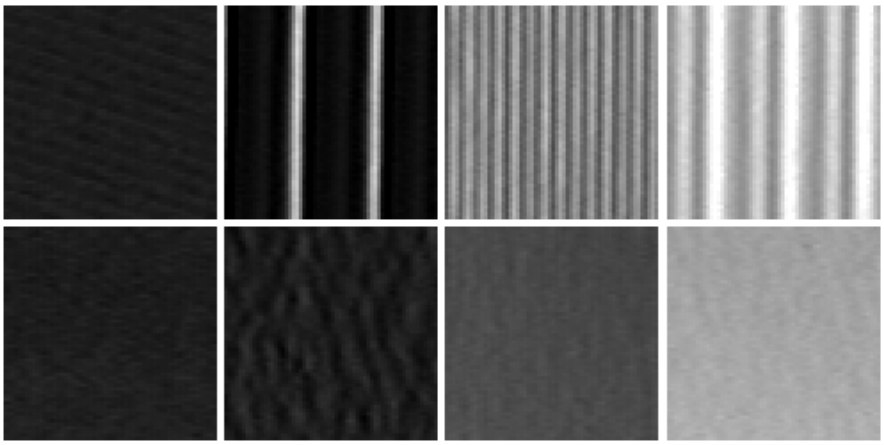


Figure 12. Models per texture: The top row shows four images of the same texture, Ribbed Paper, photographed under different viewing and lighting conditions. The images look very different. The bottom row shows images of Rough Paper taken under the same conditions as the images in the first row. These images don't differ so markedly because the texture doesn't exhibit surface normal effects. The consequence is that fewer models are required to represent Rough Paper over all viewpoints and lighting than for Ribbed Paper.

However, if we are selecting models for the express purpose of classification, then another parameter, the inter class image variation, also becomes very important in determining the number of models. For example, even if a texture varies considerably with changing imaging conditions it can be classified accurately using just a few models if all the other textures look very different from it. Conversely, if two textures look very similar then many models may be needed to distinguish between them even if they do not show much variation individually.

Broadly speaking, there are two major approaches to the problem of model reduction. In the first, various concepts from the Machine Learning literature can be used to select a subset of the models while maximising some criteria of classification and generalisation. The second approach is geometric and focuses on building descriptors invariant to imaging conditions so as to reduce the number of models needed.

3.1. Model selection

Many Machine Learning techniques have been developed to reduce the number of models in a classification algorithm. One of the simplest examples (Duda et al., 2001), for a nearest neighbour classifier, is to remove each model for which all the neighbouring models belong to the same class. This can be done safely as these models make no contribution in determining the classification boundaries (as can be seen from the Voronoi tessellation). However, in practise this has often been found not to lead to a substantial reduction in the number of models. It is also possible to reduce the number of models by completely switching classifiers. For instance, Support Vector Machines (Cristianini and Shawe-Taylor, 2000; Hayman et al., 2004; Kim et al., 2002; Scholkopf and Smola, 2002), and perhaps more appropriately Relevance Vector Machines (Tipping, 2001), are both capable of reducing the number of models while providing good generalisation.

In this subsection, we investigate two schemes for model reduction in a nearest neighbour classifier framework. Both these schemes take into account the inter and intra class image variation. Two types of experiments are performed for either method. In the first, models are selected only from the training set and classification results reported only on the test set. In the second type, the classification experiments are modified slightly so as to maximise the total number of images classified. Following (Cula and Dana, 2004), if only M models per texture are used for training, then the rest of the $46 - M$ training images are added to the 46 test images so that a total of $92 - M$ images are classified per material. For example, when classifying 61 textures, if only $M = 10$ models are used on average then a total of 82 images per texture are classified giving a total of $82 \times 61 = 5002$ test images. This is done so as to be able to make accurate comparisons with (Cula and Dana, 2004). The texton dictionary used in all experiments is the same as the one in the previous section and has 200 textons.

3.1.1. *K-Medoid algorithm*

Each histogram may be thought of as a point in $\mathbb{R}^N$, where N is the number of bins in the histogram, so that the models for a particular texture class simply consist of a set of points in $\mathbb{R}^N$ space. Given a distance function between two points, in our case χ^2, the set of points corresponding to a texture's models may be *clustered* into representative centres, and the set of points then replaced by the centres. There are many choices that can be made at this point, for example whether to cluster only within a texture class, or to take into account other classes when clustering, or to cluster the histograms of *all* the training

images irrespective of class (i.e. all the training images taken from all the texture classes). Here we only investigate the last case.

The clustering is implemented using the *K-Medoid* algorithm. This is a standard clustering algorithm (Kaufman and Rousseeuw, 1990) where the update rule always moves the cluster centre to the nearest data point in the cluster, but does not merge the points as in the case of the more popular *K-Means*. The *K-Means* algorithm can only be applied to points within a texture class. It can not be applied across classes as it merges data points and thus the resultant cluster centres can not be identified uniquely with individual textures. This is not a problem with the *K-Medoid* algorithm as the cluster centres are always data points themselves. Table II lists the results of classifying 20 textures using the four different filter banks with $K = 60, 120$ and 180, resulting in an average of $3, 6$ and 9 models per texture.

	Average # of models per texture			Average # of models per texture		
filters	3	6	9	3	6	9
S	77.47%	86.05%	91.08%	75.87%	85.76%	90.65%
LM	75.28%	85.06%	89.52%	74.89%	85.22%	89.35%
MR4	71.07%	80.93%	86.39%	71.09%	81.85%	84.57%
MR8	77.08%	89.88%	93.55%	79.35%	89.57%	93.59%
	(a)			(b)		

Table II. Classification results for each of the four filter sets when the models are automatically selected by the *K-Medoid* algorithm. In (a), the training and test sets are kept distinct while in (b) the images from the training set which are not selected as models are added to the test set and classified. Both types of experiments give very similar results, even though many more images have to be classified correctly in (b) to achieve the same performance as in (a). In all cases a dictionary of 200 textons is used and there are 20 textures being classified.

For MR8, the classification rate with 9 *K-Medoid* selected models per texture is almost as good as the 97.83% obtained using all 46 models (see column 1 in table I). In the first type of experiment (table IIa) an accuracy of 93.55% is achieved while the second type (table IIb) obtains an accuracy of 93.59% while classifying many more test images. However, clustering does have the disadvantage that very similar models are aggregated into a single cluster even if they come from different texture classes. Similarly, many clusters centres, rather than just one, might be used to represent models which are spread apart even if they belong to the same texture class. Both these shortcomings can be overcome by using a greedy algorithm which prunes the list of models on the basis of classification boundaries.

3.1.2. *Greedy algorithm*

An alternative to the *K-Medoid* clustering algorithm is a greedy algorithm, based on the post-processing step of the reduced nearest neighbour rule (Gates, 1972; Toussaint, 2002), designed to maximise the classification accuracy while minimising the number of models used. The algorithm is initialised by setting the number of models equal to the number of training images available. Then, at each iteration step, one model is discarded. This model is chosen to be the one for which the classification accuracy decreases the least when it is dropped. This iteration is repeated until no more models are left. Note that while the algorithm is constrained to select models only from the training set, classification performance is being assessed on the test set. This emulates the setup of (Cula and Dana, 2004) where the model reduction algorithm has access to both training and test images for each texture class and should therefore facilitate a faithful comparison with their work. However, it must be emphasised that in real world classification, the test set is not available for inspection to the training set and in such situations it is preferable to subdivide the training set further into model learning and validation sets.

	Average # of models per texture			Average # of models per texture		
filters	3	6	9	3	6	9
S	88.80%	96.30%	96.30%	88.37%	97.21%	98.01%
LM	87.28%	96.09%	96.20%	86.69%	95.99%	97.83%
MR4	85.22%	94.02%	94.24%	85.00%	93.66%	96.39%
MR8	93.70%	97.83%	97.83%	90.28%	98.14%	98.80%
	(a)			(b)		

Table III. Classification rates for each of the four filter sets when the models are automatically selected by the Greedy algorithm. In (a), the test set is kept distinct by not adding discarded models to it while in (b) the discarded models are added to the test set and classified. A dictionary of 200 textons is used in all cases and there are 20 textures being classified.

Table III lists the results of classifying 20 textures using the four different filter banks. It is very interesting to note that the classification accuracy obtained using 9 models can actually be better than that obtained using all 46 models (see column 1 in table I). In table IIIa, this implies that using a fewer number of models can improve performance and that the greedy algorithm is good at rejecting noisy or outlier models. In table IIIb, this also indicates that most of the training images being added to the test set are being classified correctly.

Figure 13 shows the resultant classification accuracy versus number of models for the four filter banks when classifying 20, 40 and 61 textures. For MR8, a very respectable classification rate of over 97% correct is achieved using on an average only 9 models per texture, even when all 61 classes are included. Figure 14 shows the 9 textures that were assigned the most models as well as the 9 textures that were assigned the least models while classifying all 61 textures.

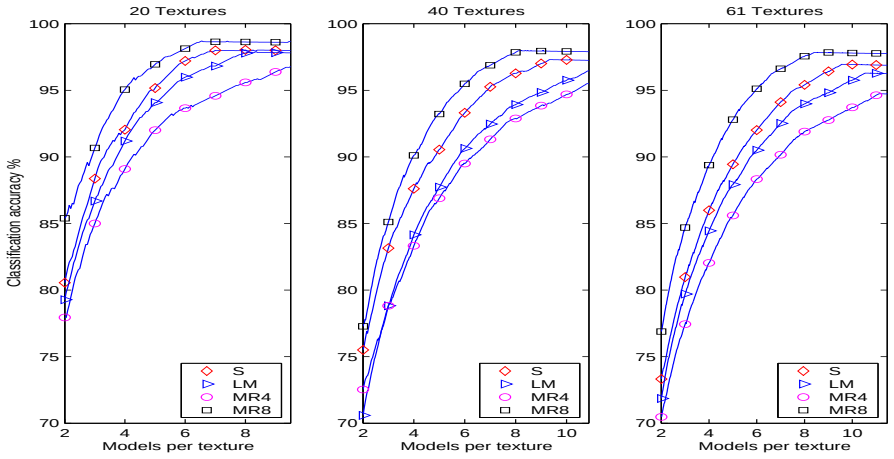


Figure 13. Classification rates for models selected by the *Greedy* algorithm for 20, 40 and 61 textures. In these experiments, the images from the training set which were not selected as models were added to the test set, as in table IIIb. The general ordering of the curves, in terms of decreasing classification performance, is MR8, S, LM and MR4. The trend is much the same even below 2 models per texture class though the graphs have been truncated for visualisation purposes. However, the ordering can sometimes change as is seen in the case while classifying 40 textures when LM slips below MR4 at between 2 and 4 models.

3.1.3. *Discussion*

The results for both the *K-Medoid* and the *Greedy* algorithms, while using the MR8 filter bank, compare very favourably with those reported in (Cula and Dana, 2004) and (Leung and Malik, 2001). In the case where there are 20 textures to be classified, the *K-Medoid* algorithm has a classification accuracy of 93.59% while using, on average, 9 models per texture class while the *Greedy* algorithm achieves an accuracy of 98.80%. In contrast, for the same 20 textures, Cula and Dana obtain a classification rate of 71% while using 8 models per texture class (by taking the most *significant* image from each texture and using a *manifold merging procedure*). This increases marginally to 72% if 11 models are used per texture (see figure 19b and table 4 in (Cula and Dana, 2004)). Note that the comparison is not exact since we classify only

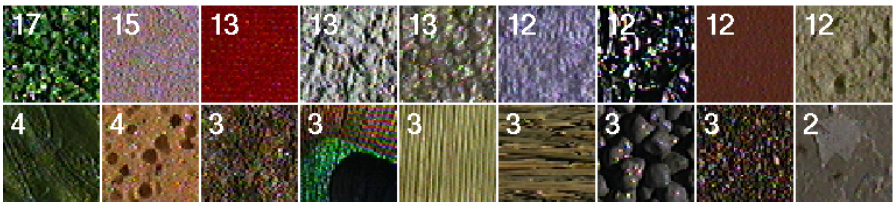


Figure 14. Models selected by the *Greedy* algorithm while classifying all 61 textures: The top row shows the 9 texture classes, and the corresponding number of models, that were assigned the most number of models by the *Greedy* algorithm while the bottom row shows the 9 classes that were assigned the least number of models. Moving from left to right, the textures and the number of models assigned to it are: Artificial grass (17), Sandpaper (15), Velvet (13), Plaster B (13), Rug A (13), Terrycloth (12), Aluminium Foil (12), Quarry Tile (12), White Bread (12), Lettuce Leaf (4), Sponge (4), Cracker A (3), Peacock Feather (3), Corn Husk (3), Straw (3), Painted Spheres (3), Roof Shingle (3) and Limestone (2).

$92 - 9 = 83$ images per texture class as compared to the $156 - \{8, 11\}$ classified by Cula and Dana. Hence, (Cula and Dana, 2004) classify many more images, some of which might be quite hard to categorise correctly because of the oblique viewing angle.

Nevertheless, there is a significant level of difference between the performance of the *K-Medoid* and the *Greedy* algorithms on one hand and the manifold method of (Cula and Dana, 2004) on the other. This is primarily due to the fact that the methods developed here take into account both the *inter* class variation, as well as intra class variation. The models that Cula and Dana learn are general models and not geared specifically towards classification. They ignore the inter class variability between textures and concentrate only on the intra class variability. The models for a texture are selected by first projecting all the training and test images into a low dimensional space using PCA. A manifold is fitted to these projected points, and then reduced by systematically discarding those points which least affect the "shape" of the manifold. The points which are left in the end correspond to the model images that define the texture. Since the models for a texture are chosen in isolation from the other textures, their algorithm ignores the inter class variation between textures.

For 40 textures, Leung and Malik report an accuracy rate of 95.6% for classifying multiple (20) images using, in effect, 20 models per texture class. For single image classification under *known* imaging conditions, using 4 models per texture class results in a drop in the accuracy rate to 87% (as computed for 5 test images per texture). The MR8 filter bank achieves 95.6% accuracy on the same textures using only 5.9 models per texture, and furthermore achieves 98.06% accuracy using, on average, 8.25 models per texture.

3.2. Pose normalisation

In this subsection we discuss some geometric approaches to model reduction. In theory, these approaches are valid only in the absence of 3D effects, i.e. for planar textures where illumination does not play a major role, and where a 3D rotation and translation of the texture is equivalent to an affine transformation of its image. However, in practise, these methods are quite robust.

The fundamental idea is to incorporate some level of geometric invariance into a model. This will ultimately allow us to be invariant to changes in the camera viewpoint and thereby reduce the number of models required to characterise a texture. The use of rotationally invariant filters is already a first step in this direction but the problem of scale still needs to be resolved (we are ignoring perspective effects for the moment). One approach could be to extend the MR sets to take the maximum response not only over all orientations but over all scales or over all affine transformations of the basic filter, but that is not investigated here. Instead we investigate the method of pose normalisation.

In (Schaffalitzky and Zisserman, 2001) it was demonstrated that, provided a texture has sufficient directional variation, it can be pose normalised by maximising the isotropy of its gradient second moment matrix (a method originally suggested in (Lindeberg and Gårding, 1994)). The method is applicable in the absence of 3D texture effects. Here we investigate if this normalisation can be used to at least reduce the effects of changing viewpoint, and hence provide tighter clusters of the filter responses, or better still reduce the number of models needed to account for viewpoint change.

In detail, if the normalisation is successful, then for moderate changes in the viewing angle, two such "pose normalised" images of the same texture should differ from each other by only a similarity transformation. If there are no major 3D scale effects, the responses of a rotationally invariant filter bank (MR or S) to these images should be much the same. A preliminary investigation shows that this is indeed the case for suitable textures.

Figure 15 shows results for two textures - Plaster A and Rough Plastic. Twelve images of each texture are selected to have similar photometric appearance (i.e. constant illumination conditions), but monotonically varying viewing angle. The graph shows the χ^2 distance between the texton histogram of one of the images (selected as the model image) and the rest, before and after pose normalisation. As can be seen, the χ^2 distance is reduced for the pose normalised images. This in turn translates to better classification as well. On experiments on 4

textures, using the same 12 image set and one model per texture, the classification rate increased from 81.81% before pose normalization to 93.18% afterwards.

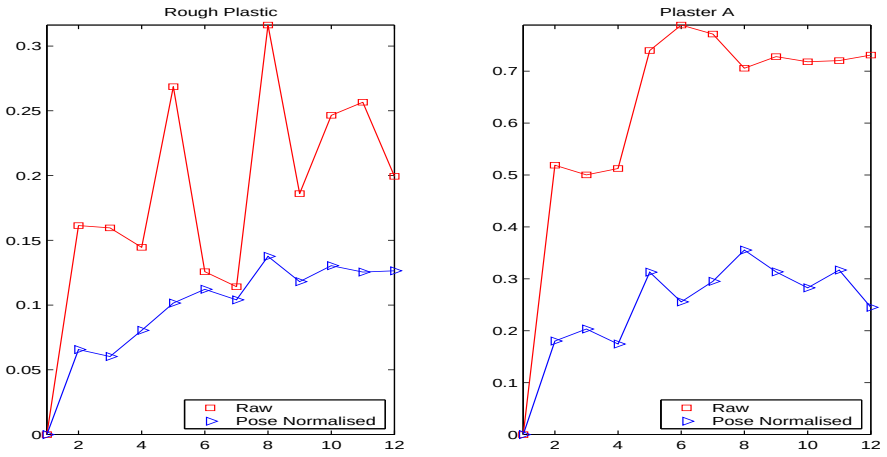


Figure 15. The effect of pose normalisation on a set of 12 images for two textures: Rough Plastic and Plaster A. The 12 images have been sorted according to increasing viewing angle and this is represented on the X axis. The Y axis is the χ^2 distance between the model image and the given image. The pose normalised images consistently have a reduced χ^2 distance which translates into better classification.

One drawback of this method is that the proposed normalisation is global rather than local. Not only would local normalisation be more robust but it would also allow the method to be extended to textures which are not globally planar but which can be approximated as being locally planar. Realising this, (Lazebnik et al., 2003b; Lazebnik et al., 2003a) proposed alternative methods of generating local, affine invariant, texture features. In their framework, certain interest regions are first detected in texture images using the Laplacian and Harris detectors. Each of these regions is then scale and pose normalised locally. Spin images are then used instead of filter banks to generate rotationally invariant features for each region. Their results are very encouraging though no direct comparison is possible as their experiments are not carried out on the CUReT database. One point of concern however, is the reliance on the detection of blob like interest regions as there exist many textures which do not exhibit such markings.

4. Generalisations

In this section, we investigate the various generalisations and modifications that can be made to the basic classification algorithm. In subsection 4.1, we study the effect of some of the more important parameters on our classifier. In particular, the effect of the choice of texton dictionary and training images is investigated. We also look at how scaling the images impacts performance. Finally, the issue of whether information is lost by using just the first order statistics of rotationally invariant filter responses is discussed in section 4.2. A method for reliably measuring relative orientation texton co-occurrence is presented in order to incorporate second order statistics into the classification scheme.

4.1. Algorithm parameter variations and the issue of scale

In this subsection, various parameters of the algorithm are varied and the effect on the classification performance determined. We first calculate a benchmark classification rate and then vary the images in the training set and also the size of the texton dictionary to see how performance is affected.

For the benchmark case, the texton dictionary is built by learning 10 textons from each of the 61 textures (using the procedure described in subsection 2.3) to have a total of 610 textons. The 46 training images per texture from which the models will be generated are chosen by selecting every alternate image from the set of 92 available. Under these conditions, the MR8 filter bank achieves a classification accuracy of 96.93% using 46 models per texture for all 61 textures. On running the greedy algorithm the classification accuracy increases to 98.3% using, on average, only 8 models per texture. This defines the benchmark rate.

We now investigate the effect of choice of textons on the classification performance. First we reduce the number of textons by learning 10 textons from only 31 randomly chosen textures to get a dictionary of 310 textons, and then repeat the experiment of section 2. The classification rate decreased only slightly from the benchmark to 98.19%.

The number of textons in the dictionary can be further reduced by merging textons which lie very close to each other in filter response space. The texton dictionary can be pruned down from 310 to 100 by selecting 80 of the most distinct textons (i.e. those textons that didn't have any other textons lying close by) and then running *K-Means*, with $K = 20$, on the rest. This procedure entailed another slight decrease in the classification accuracy to 97.38%. These results indicate that the pruned dictionaries are still universal (Leung and Malik, 2001), i.e.

texton primitives learnt from some randomly chosen texture classes can be used to successfully characterise other classes as well.

We now increase the size of the texton dictionary to see if classification improves accordingly. Table IV gives a summary of the results. The best performance is obtained with a dictionary of 2440 textons when the classification accuracy is 97.43% using 46 models per texture. On running the greedy algorithm, the number of models used is reduced to, on average, 7.14 per texture. If the unused training images are added to the test set, the classification rate improves to 98.61%.

Essentially we are comparing different representations of the joint probability distribution of filter responses in terms of their classification performance. A set of textons can be thought of as adaptively partitioning the space of filter responses into bins (determined by the Voronoi diagram) and a histogram of texton frequencies can be equated to a probability distribution over filter responses (Varma and Zisserman, 2005). In such a situation, the number of bins should not be too few otherwise the approximation to the true PDF will be poor nor should there be too many bins so as to prevent over-fitting.

Number of	Before Greedy		After Greedy	
Textons	Classification	Models	Classification	Models
1220	97.11%	46	98.43%	7.56
1830	97.18%	46	98.49%	7.26
2440	97.43%	46	98.61%	7.14
3050	97.32%	46	98.57%	7.41

Table IV. The effect of increasing the size of the texton dictionary while classifying all 61 textures from the CUReT database.

As can be seen in table IV there is a point beyond which increasing the number of textons actually decreases performance as the data is now being over fitted. This can be used to automatically select the appropriate number of textons for a given problem by partitioning the data into a training and validation set and then choosing the texton dictionary which maximises classification on the validation set.

We now turn to the choice of training images. It could be argued that the results presented here are biased as the training set has been chosen by including every alternate image from the set of 92 available per texture. We address this issue by repeating the classification experiment but with the training images chosen randomly. The dictionary of 2440 textons generated previously is used and the experiment repeated 50,000 times. Figure 16 shows the distribution of classification results when 46 images were chosen randomly from every texture class to form

Training images per texture	Classification Statistics			
	Mean	STD	Min	Max
46	97.28%	0.316%	95.72%	98.40%
23	94.22%	0.456%	91.97%	95.82%
12	89.02%	0.679%	85.92%	91.84%
6	80.67%	0.986%	76.46%	84.50%
3	69.70%	1.373%	63.90%	75.52%

Table V. Classification statistics when the training images were chosen randomly. A dictionary of 2440 textons was used and all 61 textures were classified. In each case, the statistics were gathered over 50,000 runs of the classification experiment.

the training set while table V provides a summary of the results for varying sizes of the training set. The mean classification accuracy when the 46 models were chosen randomly was 97.28% which is very similar to the 97.43% obtained when the 46 images were chosen by including every alternate image. This shows that our experimental setup is not biased and that we are not over fitting to the data.

In summary, the best classification rate achieved, while classifying all 61 textures, was 98.61% obtained when 2440 textons were used and the worst rate was 97.38% when only 100 textons were used. These results are listed in table VI. We can therefore conclude that our algorithm is

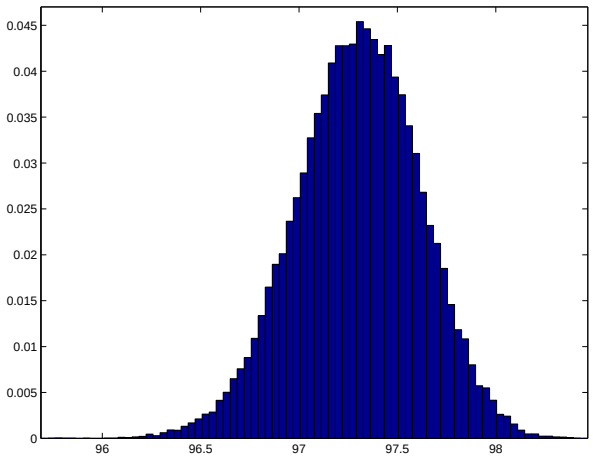


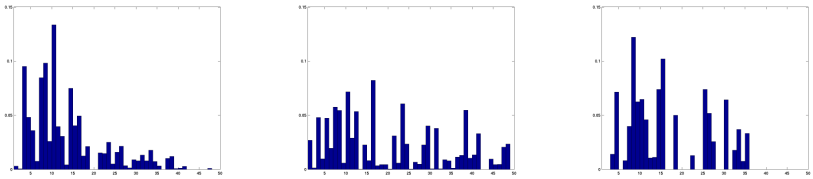
Figure 16. The distribution of classification percentages when 46 training images are chosen randomly per texture from the set of 92 available. The experiment was run 50,000 times with a dictionary of 2440 textons and all 61 materials in the CUReT database were classified. The mean classification accuracy was 97.28% with a standard deviation of 0.316%. The maximum was 98.4% and the minimum was 95.72%.

robust and relatively insensitive to the choice of training image set and texton vocabulary with the classification rate not being affected much by changes in these parameters.

	Number of Textons	Before Greedy		After Greedy	
		Classification	Models	Classification	Models
Worst	100	95.32%	46	97.38%	9.83
Benchmark	610	96.93%	46	98.30%	8.00
Best	2440	97.43%	46	98.61%	7.14

Table VI. Benchmark, worst and best case results for varying parameters of the classification algorithm.

Finally, a word about scale. It may be of concern that the MR4 filter bank does not have filters at multiple scales and hence will be unable to handle scale changes successfully. To test this, 25 images from 14 texture classes were artificially scaled, both up and down, by a factor of 3. The classification experiment was repeated using the original, normal sized, filter banks and texton dictionaries. We found that as long as models from the scaled images were included as part of the texture class definition, classification accuracy was virtually unaffected and classification rates of over 97% were achieved. However, if the choice of models was restricted to those drawn from the original sized images, then the classification rate dropped to 17%. It is evident from this that filter bank and texton vocabulary are sufficient, and it is the model that must be extended (see figure 17).



(a) (b) (c)

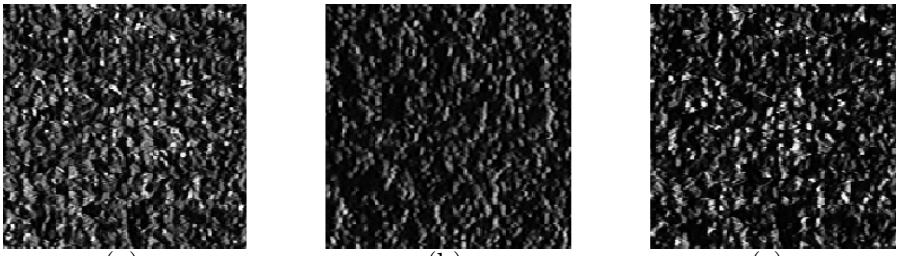
Figure 17. Scaling the data results in new models: The histogram of texton labellings of (a) the original image (b) the image scaled up by a factor of 3 and (c) the image scaled down by a factor of 3. All three models are substantially different indicating that the model must be extended.

4.2. Orientation co-occurrence

The classification scheme, up to this stage, has only used information about first order texton statistics (i.e. their frequency and not a

measure of their co-occurrence). However, recent research into texture driven content-based image retrieval (Schmid, 2001) has shown that a hierarchical system which uses co-occurrence of textons over a spatial neighbourhood can lead to good results. Therefore, in this subsection, we investigate whether incorporating such second order statistics can improve classification performance on the CUReT database.

As was seen in the previous subsection, classification on the basis of texton frequency information alone is already very good and rates of over 97% can be achieved. What is also interesting is that, of the images that were misclassified, the correct texture class was ranked within the top 5 most of the times. Figure 18 shows how similar one of the misclassified novel images is to both the top ranked, but incorrect, texture model and the second ranked, but correct, model. Since the MR8 filter bank is rotationally invariant, there is the possibility that some of these misclassifications are due to two different texture classes, which are not rotationally related, being mapped to the same texton frequency distribution. Therefore, we focus on the question of whether incorporating second order texton statistics, in the form of co-occurrence of angles, can improve classification (though the method developed here is general and can also be applied to spatial co-occurrence).



(a) (b) (c)

Figure 18. Misclassifications: (a) is an image of Artificial Grass taken from the test set which was misclassified as (b) Pebbles. The next closest model image to (a) is (c) which belongs to the correct texture class - Artificial Grass. The misclassified novel image is perceptually quite similar to both the correct and the incorrect model images.

4.2.1. *Reliably measuring a relative orientation co-occurrence statistic*
Given a texton in an image labelling, the objective is to measure the relative angle of occurrence of surrounding textons, that lie within a circular neighbourhood, with respect to the given texton. Certain difficulties have to be overcome in order to reliably measure this relative angle co-occurrence. Firstly, the angles of occurrence of the textons have to be measured robustly. Conventionally, working in a match filter paradigm, the orientation of a feature (such as an edge or a bar) is

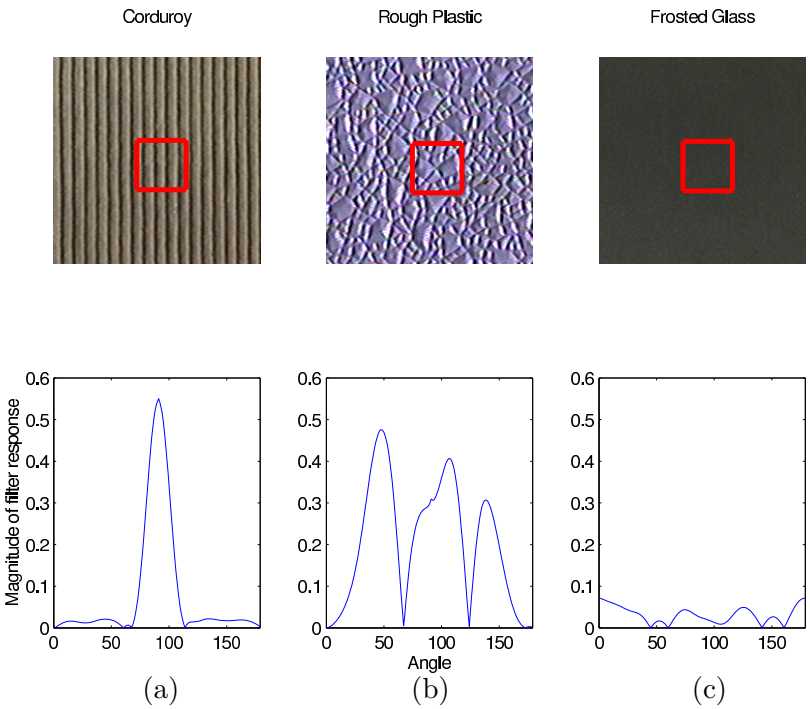


Figure 19. Determining the orientation of image features: The top row shows images of 3 textures (a) Corduroy, (b) Rough Plastic and (c) Frosted Glass, with a highlighted central image patch which is matched with an edge filter at all orientations. The magnitude of the filter response versus the orientation is plotted in the bottom row. As can be seen: (a) is a strongly oriented texture having a single direction and therefore its filter response is uni-modal; (b) the texture contains edges along several directions and this is reflected in its filter response; (c) the texture is isotropic and the features have no specific orientation. Plots (b) and (c) show that defining the orientation of a feature to be the angle at which the maximal filter response occurs can be unstable.

determined to be the angle of maximum response of a filter designed to match that feature. However, features can occur at multiple angles at the same point and, as such, it is difficult to assign them a particular orientation (see figure 19). For instance, an edge filter will have a maximal response at two orientations when matching a corner and choosing one edge orientation over the other will lead to instabilities. Note that these instabilities do not affect the MR representation because only the value of the response (not its angle) is significant — if the same value occurs at two orientations the orientation corresponding to the maximum response is unstable, but the maximum response is not. Here we use the orientated filter (of MR8) that has the maximum response to determine the orientation.

Returning to relative orientation, a robust representation can be obtained if the magnitude of the filter response at each angle (normalised so that the sum of magnitudes squared over all angles is unity) is treated as a confidence measure in the feature occurring at that orientation. Thus, in our case, this *normalised magnitude vector* will be a 6 vector representing the confidence that the given feature occurs at the 6 angles corresponding to the orientations present in the MR8 filter bank (though a richer representation can be obtained using approximated steerable kernels and interpolation (Perona, 1992)). The relative angles between two features, which is invariant to rotation, can now be calculated by computing the cross-correlation between their normalised magnitude vectors. Given a central texton, we can compute the frequency with which other textons occur at various relative angles to it by forming the sum of the cross-correlations between the normalised magnitude vectors of the central texton and the surrounding textons. Essentially, this is computing (via soft binning) the count of how many times a neighbouring texton occurs at a given angle relative to the central texton. To maintain rotational invariance, the surrounding textons come from a circular neighbourhood with a predefined radius, centred around the given texton.

4.2.2. *Extending the classification algorithm*

Now that a co-occurrence 6-vector can be associated with every texton in an image labelling, the classification algorithm can be extended to use the joint distribution of filter responses and co-occurrence vectors. Just as filter responses were clustered into filter response textons in section 2.3, co-occurrence vectors can be clustered to find exemplars as well, and a dictionary of co-occurrence vector textons can be formed. Textons from this dictionary can be used to label the co-occurrence vectors for a given image. The model for a training image then becomes the joint histogram of the frequency of occurrence of filter response textons and co-occurrence vector textons. Thus, a model is an $K_{fr} \times K_{cv}$ matrix M where K_{fr} is the number of filter response textons and K_{cv} is the number of co-occurrence vector textons. Each entry M_{ij} in this matrix represents the probability of filter response texton K_{fr_i} and orientation co-occurrence texton K_{cv_j} occurring together in the training image. This is somewhat similar to the co-occurrence representation of (Schmid, 2001). To classify a novel image, its joint histogram is built and is then compared to all the models using χ^2 over all elements of the M matrix. Thus, the essence of the classifier remains the same, the only extension is that joint distribution of filter response and co-occurrence textons are used rather than just the histogram of filter

response textons. Hence, we get to add extra information and yet retain all the benefits of our existing classification scheme.

4.2.3. *Experimental Setup and Results*
The orientation co-occurrence texton dictionary is created by clustering the co-occurrence vectors (calculated for a particular radius of the circular neighbourhood) from the same set of 13 training images per texture that were used to generate the filter response texton dictionary. The filter responses and co-occurrence vectors of the training images are then labelled using the two texton dictionaries. Finally, the models are built by forming the frequencies, in the $K_{fr} \times K_{cv}$ texton space, of the joint occurrence of the filter response textons and the orientation co-occurrence textons.

Obviously, the choice of K_{fr} and K_{cv} is important as $K_{fr} \times K_{cv}$ equals the number of bins and therefore determines how accurately the joint PDF is approximated. However, we cannot choose $K_{fr} = 610$ as had been done previously, because the number of bins becomes too large and we start over-fitting the data (see table VII (a)-(c)). A lower value, such as $K_{fr} = 30$, was found to be more appropriate. Table VII (d)-(f) lists the classification results obtained for various values of the radius when K_{cv} is also set to 30. The performance, using the joint representation, is better than using just 30 filter response textons or just 30 co-occurrence vector textons. Though it is worse than if 900 filter

Radius	610 FR Textons (a)	610 CV Textons (b)	610×610 Joint Textons (c)	30 FR Textons (d)	30 CV Textons (e)	30×30 Joint Textons (f)
01	96.86%	74.51%	88.02%	92.94%	63.93%	95.22%
02	96.75%	68.13%	85.28%	92.62%	60.08%	94.72%
05	96.86%	65.39%	85.88%	92.87%	54.84%	94.15%
10	96.65%	61.26%	85.13%	92.23%	48.68%	93.33%

Table VII. Classification results for all 61 textures using 46 models per texture when orientation co-occurrence information is incorporated into the classification scheme. (a) classification accuracy if only 610 filter response (FR) textons are used to label images and build models. There are minor variations in the classification rate as the number of points available for labelling changes with the radius. (b) classification accuracy if only 610 co-occurrence vector (CV) textons are used. (c) classification rate if the joint distribution is used. The results are poor as there are too many bins and the data is being over fitted. The next three columns have the same format except now both the texton dictionaries have been pruned to 30 textons each. The joint classification rate improves and is better than either of the marginals, though it is still not as good as that obtained by just using 900 FR textons.

response textons were used without any co-occurrence. If the radius is kept fixed and K_{cv} varied then the performance of the joint representation, predictably, first increases, reaches a maximum and then falls (though in no case is it ever able to surpass the performance achieved using an equivalent number of filter response textons alone).

These results indicate, that at least for this dataset, the density of filter response textons is the best measure of discrimination and that orientation co-occurrence does not help much in classification (similar results were found for spatial co-occurrence as well). They also confirm that rotational invariance is advantageous and that no significant information is being lost in this case by using a rotationally invariant filter bank.

5. Conclusions

In this paper, we have tackled the problem of texture classification and have demonstrated how single images can be classified using a few models without requiring any information about their imaging conditions. This is a substantial improvement over previous work which required multiple images obtained under known conditions. We have also introduced rotationally invariant, low dimensional, maximum response filter banks which were shown to have superior performance as compared to traditional filters due to enhanced feature detection and clustering. Moreover, we presented two novel methods for reducing the number of models needed to characterise textures and again demonstrated their superiority over existing algorithms. It was also shown that the proposed classification scheme is robust to the choice of training images and texton dictionaries. Finally, we concluded that even though the classifier can be extended by incorporating second order statistics this does not lead to an improvement in the overall classification. This implies that using only the frequency distribution of textons is sufficient and that no significant information is being lost by employing rotationally invariant filter banks for this database.

This research has benefited greatly from the availability of the Columbia-Utrecht database. The CUReT database is a considerable improvement over the previously used Brodatz collection (Brodatz, 1966), though it also has some limitations. Its main advantages are that it has many real world textures photographed under varying image conditions, and the effects of specularities, shadowing and other surface normal variations are evident. The limitations of the CUReT database are mainly in 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 textures

and limited in-plane rotation. As regards choice of texture, the most serious drawback is that multiple instances of the same texture are present for only a very few of the materials, so intra-class variation cannot be investigated. Hence, it is difficult to make generalisations.

The time is now right for a yet more demanding database which overcomes the above limitations, and also includes non-planar surfaces.

Acknowledgements

We are grateful to Oana Cula, Tomas Leung and Cordelia Schmid for supplying details of the algorithms used in their papers. We are also grateful to Frederik Schaffalitzky for numerous discussions and suggestions. Financial support was provided by a University of Oxford Graduate Scholarship in Engineering at Jesus College, an ORS award and the EC project CogViSys.

References

Brodatz, P.: 1966, *Textures: A Photographic Album for Artists & Designers.* New York: Dover.

Chantler, M. J., G. McGunnigle, A. Penirschke, and M. Petrou: 2002a, 'Estimating Lighting Direction and Classifying Textures'. In: *Proceedings of the 13th British Machine Vision Conference, Cardiff.* pp. 737–746.

Chantler, M. J., G. McGunnigle, and J. Wu: 2000, 'Surface rotation invariant texture classification using photometric stereo and surface magnitude spectra'. In: *Proceedings of the 11th British Machine Vision Conference, Bristol.* pp. 486–495.

Chantler, M. J., M. Schmidt, M. Petrou, and G. McGunnigle: 2002b, 'The Effect of Illuminant Rotation on Texture Filters: Lissajous's Ellipses'. In: *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark.* pp. 289–303.

Cristianini, N. and J. Shawe-Taylor: 2000, *An Introduction to Support Vector Machines.* Cambridge University Press.

Cula, O. G. and K. J. Dana: 2004, '3D Texture Recognition Using Bidirectional Feature Histograms'. *International Journal of Computer Vision* **59**(1).

Dana, K. J. and S. Nayar: 1998, 'Histogram model for 3d textures'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* pp. 618–624.

Dana, K. J. and S. Nayar: 1999, 'Correlation Model for 3D Texture'. In: *Proceedings of the International Conference on Computer Vision.* pp. 1061–1067.

Dana, K. J., B. van Ginneken, S. K. Nayar, and J. J. Koenderink: 1999, 'Reflectance and Texture of Real World Surfaces'. *ACM Transactions on Graphics* **18**(1), 1–34.

Duda, R. O., P. E. Hart, and D. G. Stork: 2001, *Pattern Classification.* John Wiley and Sons, 2nd edition.

Fowlkes, C., D. Martin, X. Ren, and J. Malik: 2002, 'Detecting and Localizing Boundaries in Natural Images'. Technical report, University of California at Berkeley.

Funt, B., K. Barnard, and L. Martin: 1998, 'Is colour constancy good enough?'. In: *Proceedings of the European Conference on Computer Vision*. pp. 445–459, Springer-Verlag.

Gates, G. W.: 1972, 'The Reduced Nearest Neighbour Rule'. *IEEE Transactions on Information Theory* **18**(3), 431–433.

Hayman, E., B. Caputo, M. Fritz, and J. Eklundh: to appear in 2004, 'On the Significance of Real-World Conditions for Material Classification'. In: *Proceedings of the European Conference on Computer Vision*.

Kaufman, L. and P. J. Rousseeuw: 1990, *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, NY, USA.

Kim, K. I., K. Jung, S. H. Park, and H. J. Kim: 2002, 'Support Vector Machines for Texture Classification'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(11), 1542–1550.

Konishi, S. and A. L. Yuille: 2000, 'Statistical cues for Domain Specific Image Segmentation with Performance Analysis'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 125–132.

Lazebnik, S., C. Schmid, and J. Ponce: 2003a, 'Affine-invariant local descriptors and neighborhood statistics for texture recognition'. In: *Proceedings of the International Conference on Computer Vision*. pp. 649–655.

Lazebnik, S., C. Schmid, and J. Ponce: 2003b, 'A Sparse Texture Representation Using Affine-Invariant Regions'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2. pp. 319–324.

Leung, T. and J. Malik: 2001, 'Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons'. *International Journal of Computer Vision* **43**(1), 29–44.

Lindeberg, T. and J. Gårding: 1994, 'Shape-Adapted Smoothing in Estimation of 3-D Depth Cues from Affine Distortions of Local 2-D Brightness Structure'. In: *Proceedings of the 3rd European Conference on Computer Vision, Stockholm, Sweden*. pp. 389–400.

Malik, J., S. Belongie, T. Leung, and J. Shi: 2001, 'Contour and Texture Analysis for Image Segmentation'. *International Journal of Computer Vision* **43**(1), 7–27.

Penirschke, A., M. J. Chantler, and M. Petrou: 2002, 'Illuminant Rotation Invariant Classification of 3D Surface Textures using Lissajous's Ellipses'. In: *Proceedings of the 2nd International Workshop on Texture Analysis and Synthesis, Copenhagen, Denmark*. pp. 103–108.

Perona, P.: 1992, 'Steerable-Scalable Kernels for Edge Detection and Junction Analysis'. In: *European Conference on Computer Vision*. pp. 3–18.

Press, W., S. Teukolsky, W. Vetterling, and B. Flannery: 1992, *Numerical Recipes in C (2nd Ed.)*. Cambridge University Press.

Schaffalitzky, F. and A. Zisserman: 2001, 'Viewpoint Invariant Texture Matching and Wide Baseline Stereo'. In: *Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada*. pp. 636–643.

Schmid, C.: 2001, 'Constructing models for content-based image retrieval'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2. pp. 39–45.

Scholkopf, B. and A. Smola: 2002, *Learning with Kernels*. MIT Press.

Suen, P. and G. Healey: 2000, 'The analysis and reconstruction of real-world textures in three dimensions'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(5), 491–503.

Tipping, M.: 2001, 'Sparse Bayesian learning and the relevance vector machine'. *Journal of Machine Learning Research* **1**, 211–244.

Toussaint, G.: 2002, 'Proximity graphs for nearest neighbor decision rules: recent progress'. In: *Interface 2002, 34th Symposium on Computing and Statistics*.

Varma, M. and A. Zisserman: 2002, 'Classifying Images of Materials: Achieving Viewpoint and Illumination Independence'. In: *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, Vol. 3. pp. 255–271, Springer-Verlag.

Varma, M. and A. Zisserman: to appear in 2005, 'On Unifying Statistical Texture Classification Frameworks'. *Image and Vision Computing*.