Classifying Images of Materials: Achieving Viewpoint and Illumination Independence

Manik Varma and Andrew Zisserman

Robotics Research Group
Department of Engineering Science
University of Oxford
Oxford, OX1 3PJ
{manik,az}@robots.ox.ac.uk

Abstract. In this paper we present a new approach to material classification under unknown viewpoint and illumination. Our texture model is based on the statistical distribution of clustered filter responses. However, unlike previous 3D texton representations, we use rotationally invariant filters and cluster in an extremely low dimensional space. Having built a texton dictionary, we present a novel method of classifying a single image without requiring any a priori knowledge about the viewing or illumination conditions under which it was photographed. We argue that using rotationally invariant filters while clustering in such a low dimensional space improves classification performance and demonstrate this claim with results on all 61 textures in the Columbia-Utrecht database. We then proceed to show how texture models can be further extended by compensating for viewpoint changes using weak isotropy.
The new clustering and classification methods are compared to those of Leung and Malik (ICCV 1999), Schmid (CVPR 2001) and Cula and Dana (CVPR 2001), which are the current state-of-the-art approaches.

1 Introduction

The objective of this paper is the classification of materials from their imaged appearance. Texture is a material property of which we have only an intuitive, and not a sound mathematical, understanding. Therefore, classifying materials by their textural appearance in single images photographed under unknown viewing and illumination conditions is still quite an outstanding problem, though significant progress has been made recently [2,3,9,12,13,17,20].

In particular, Leung and Malik [13] made an important innovation in 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. In the same paper Leung and Malik also made a serious attempt on the problem of classifying textures under varying viewpoint and illumination. Their solution was a 3D texton which is a cluster centre of filter responses over a stack of images with representative viewpoints and lighting.

A. Heyden et al. (Eds.): ECCV 2002, LNCS 2352, pp. 255–271, 2002.

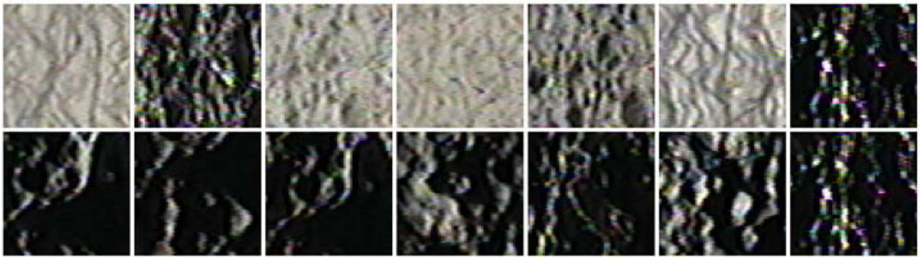

Fig. 1. The change in image appearance of the same texture (# 30, Plaster B) with variation in imaging conditions. Top row: constant viewing angle and varying illumination. Bottom row: constant illumination and varying viewing angle. There is a considerable difference in the appearance across images.

The need to seriously address "3D effects" – where the appearance varies considerably with viewpoint and lighting – is illustrated in figure 1. The importance of such effects for classification [1,4,5] and synthesis [15,19] has also been noted by other researchers.

Our approach to the classification problem is to model a texture as a distribution over textons, and learn the textons and texture models from training images. Classification of a novel image then proceeds by mapping the image to a texton distribution and comparing this distribution to the learnt models. Consequently this is quite a standard approach, but the originality is at two points: First, our texton clustering is in an extremely low dimensional space and is also rotationally invariant. The second innovation is to classify the texture from a *single* image, and we achieve this while representing each 3D texture by a small set of models.

This is a considerable difference in approach to that of [13] to which we compare our results, and it is worth elaborating on the difference at this point. The main thrust of Leung and Malik's work is towards classifying images using 3D textons. In the learning stage, 20 images of each texture are geometrically registered and mapped to a 48 dimensional filter response space. The registration is necessary because the clustering that defines the texton is in the stacked $20 \times 48 = 960$ dimensional space (i.e. the textons are 960-vectors), and it is important that each filter is applied to the same texture point as viewpoint and illumination vary.

In the classification stage, 20 novel images of the same texture are presented. However, again these images must be registered and more significantly must have the same order as the original 20 (i.e. they must be taken from images with similar viewpoint and illumination to the original). In essence, the viewpoint and lighting are being supplied implicitly by this ordering. Leung and Malik also use an MCMC algorithm for classifying a single image under *known* imaging conditions. However, the classification rate of this method is 87%, far inferior to the 95.6% achieved by the multiple image method.

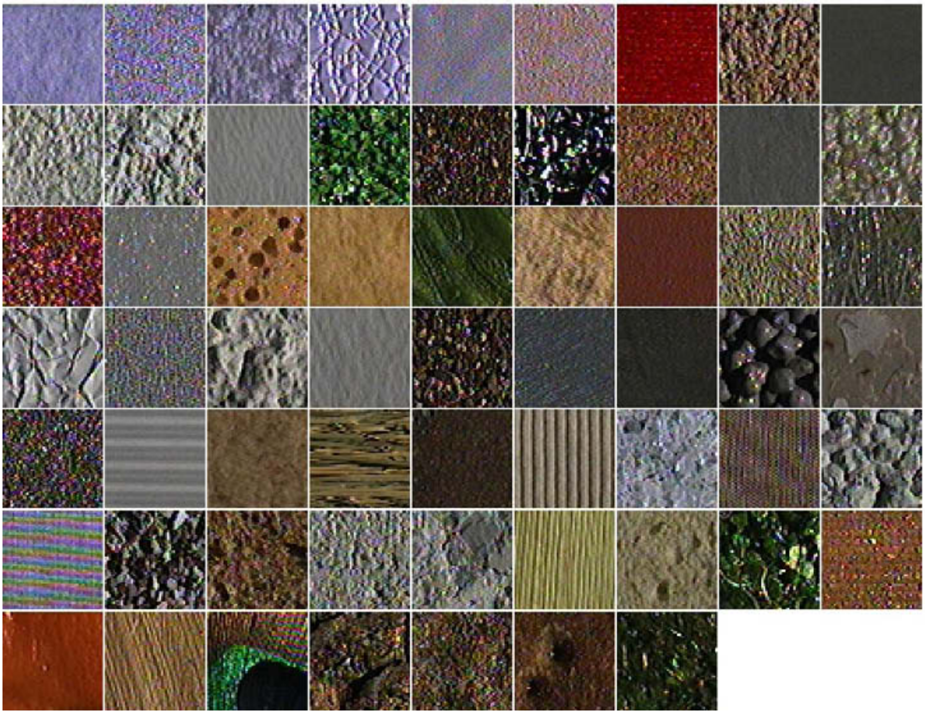

Fig. 2. Textures from the Columbia-Utrecht database. All images are converted to monochrome in this work, so colour is not used in discriminating different textures.

Here we present a texture classification method with superior performance to [13], but requiring only a single image as input and with no information (implicit or explicit) about the illumination and viewing conditions.

Our approach is most closely related to that of Schmid [18] and Cula and Dana [2,3]. Schmid's approach is rotationally invariant but the invariance is achieved in a different manner and texton clustering is in a higher dimensional space than here. Cula and Dana's approach is not rotationally invariant and the method of model selection differs from here.

The paper is organized as follows: section 2 is concerned with rotationally invariant clustering and the classification rates of four filter sets are compared. The sets include those used by Schmid [18] and Leung and Malik [13]. Then, in section 3, we describe how the number of models for each texture can be substantially reduced.

We present results on the Columbia-Utrecht database [6], the same database used by [2,13]. The 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 illustrated in figure 2.

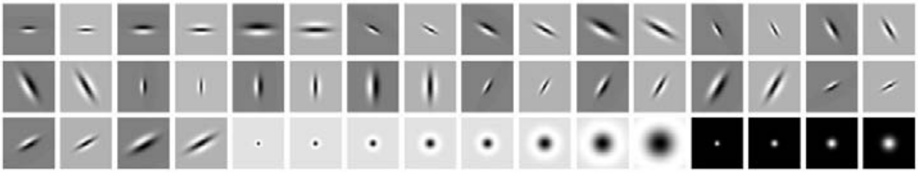

Fig. 3. The LM filter bank has a mix of edge, bar and spot filters at multiple scales and orientations. It has a total of 48 filters - 6 oriented filters at 3 scales and 2 phases, 8 Laplacian of Gaussian filters and 4 Gaussian filters.

2 Rotationally Invariant Filters and Their Effect on Textons

In this section we investigate the advantages and disadvantages of clustering in a rotationally invariant filter response space. The use of such filters has already been espoused by Schmid [18]. To be definite, we will compare four filter sets: those of Leung and Malik which are not rotationally invariant; those of Schmid which are; and two reduced sets of filters proposed here using maximum response (which are again rotationally invariant). We will assess the filter sets by the classification rate they achieve based on textons clustered in their response space.

2.1 Description of the Four Filter Sets

The Leung-Malik (LM) set: consists of 48 filters, partitioned as follows: Second derivative of Gaussians at 6 orientations, 3 scales and 2 phases (where phase refers to even and odd symmetry) making a total of 36; 8 Laplacian of Gaussian filters; and 4 Gaussians. The scale of the filters range between $\sigma = 1$ and $\sigma = 10$. They are shown in figure 3.

The Schmid (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),

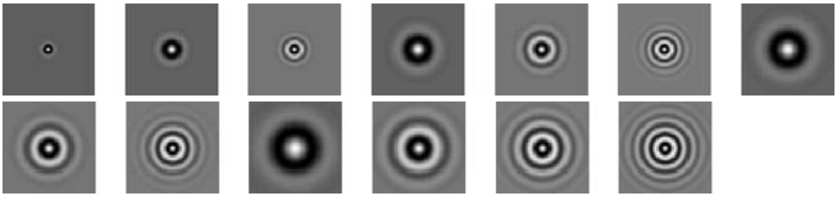

Fig. 4. The S filter bank is rotationally invariant and has 13 isotropic, "Gabor-like" filters.

(10,2), (10,3) and (10,4). The filters are shown in figure 4. As can be seen all the filters have rotational symmetry.

The Maximum Response (MR) sets: The MR4 filter bank consists of 14 filters but only four 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. In a similar spirit to [11,12] we have chosen the scale of our filters as that which gives the strongest response to the textures in the database, and only use a single scale for each filter (we return to the issue of scale below).

The filter bank is shown in figure 5 and consists of a Gaussian and a Laplacian of Gaussian both with $\sigma = 10$ (these filters have rotational symmetry), an edge filter ($\sigma_x = 4, \sigma_y = 12$), and a bar filter ($\sigma_x = 4, \sigma_y = 12$). The latter two filters are oriented. Measuring only the maximum response reduces the number of responses from 14 (six orientations for two oriented filters, plus two isotropic) to 4.

The MR8 filter bank is similar to the MR4 filter bank but the oriented edge and bar filters occur at 3 scales (σ_x,σ_y)={(1,3), (2,6), (4,12)} thereby giving a total of six responses, three for the edge filter and three for the bar. The remaining two filters are the Gaussian and Laplacian of Gaussian with $\sigma = 10$ taken from the MR4 filter bank.

The motivation for introducing these MR filters sets is twofold. First, we achieve rotation invariance, but are also able to record the angle of maximum response. This angular information is lost in the S filters. Thus, with the MR filters we are able to compute co-occurrence statistics on orientation, and such statistics may prove useful in discriminating isotropic from anisotropic textures. We return to this point in section 4. The second motivation concerns the low dimensionality of the filter response space. We would like to use the entire filter response of a texture for classification (in the manner of [12]), rather than its vector quantization into textons. Again, we return to this issue in section 4.

2.2 Pre- and Post-processing

When computing the filter responses the following three steps are followed:

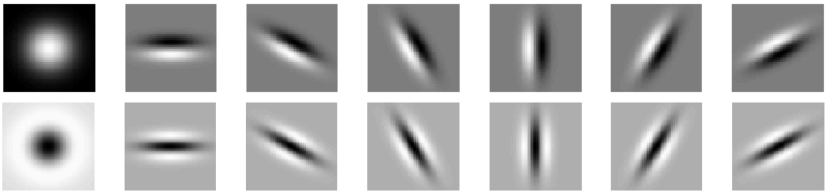

Fig. 5. The MR4 filter bank consists of four filters - two rotationally symmetric ones (a Gaussian and a Laplacian of Gaussian), and two oriented ones (an edge and a bar filter, at six different orientations).

First, before any of the filters are applied, all the images are converted to grey scale and are intensity normalised to have zero mean and unit standard deviation. This normalization gives partial invariance to linear transformations in the illumination conditions of the images.

Second, all 4 filter banks are L_1 normalised so that the filter 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.

Third, following [8] (and motivated by Weber's law), the filter response at each pixel $\mathbf{x}$ is (contrast) normalized as

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

where $L = \|\mathbf{F}(\mathbf{x})\|_2$ 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 learning from training images.

For each filter set, we adopt the following procedure for computing a texton dictionary: For each texture, we select 13 images randomly (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 [7]. The learnt textons for each texture are then aggregated into a single dictionary. For example, if 5 textures are used then the dictionary will contain 50 textons.

However, the classification accuracy achieved is not very sensitive to variations in the procedure for dictionary generation (such as the particular textures

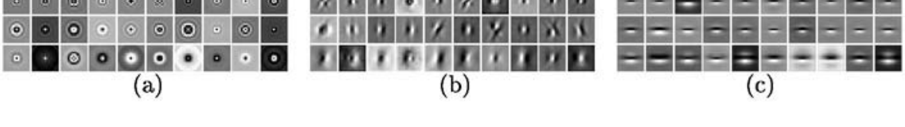

Fig. 6. The first 100 textons recovered from 20 training textures using 13 images per texture: (a) S textons. (b) LM textons. (c) MR8 textons. Note that the S textons are rotationally symmetric.

and method of choosing images etc), see subsection 3.3. Examples of the textons for each filter set are shown in figure 6.

Our clustering task is considerably simpler than that of Leung and Malik, and Cula and Dana (who use 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 a different point 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 figures 7 and 8).

This establishes that there is an advantage in rotation invariance in that rotated versions of the same texture can be represented by one histogram, whilst 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.

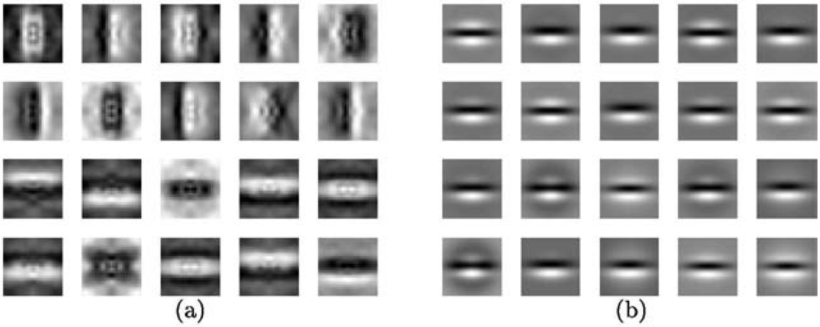

Fig. 7. Textons learnt from original and rotated textures: (a) textons learnt from the LM filter bank for an original texture image of ribbed paper and a version rotated by 90 degrees (shown in figure 8). (b) depicts the textons learnt by the MR4 filter bank. In each case, the 10 textons learnt from the original image are shown in the top 2 rows while the 10 learnt from the rotated image are shown in the bottom 2 rows. Note that the LM filter bank learns two sets of textons (where one set is essentially the other set rotated by 90 degrees) and therefore will have two very different histograms for the same texture. Conversely, the MR4 textons are virtually identical.

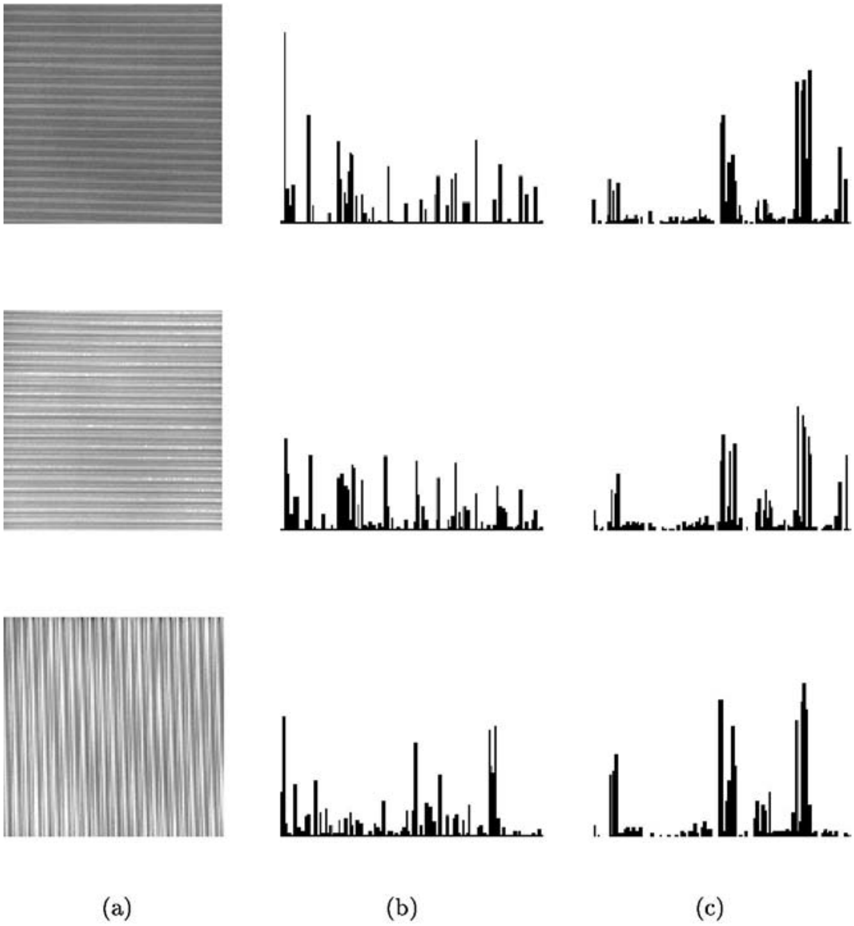

Fig. 8. Classification of rotated textures. Column (a) shows three images of the ribbed paper texture (the first two are from # 38B, the last from # 38). Column (b) shows the texton histograms using the LM filter bank. Column (c) shows the texton histograms obtained using the MR filter bank. Note that in the LM case, neither of the top two histograms matches the third (as seen through the eyes of the χ^2 distance where small values in a bin are significant), i.e. the description is rotationally variant. In the MR case all the histograms are, more or less, similar.

2.4 Classification Method and Results

Here we perform three experiments to assess texture classification rates over 92 images for each of 20, 40 and 61 textures respectively. The first experiment, where we classify images from 20 textures, corresponds to the setup employed by Cula and Dana [2]. The second experiment, where 40 textures are classified, is modeled on the setup of Leung and Malik [13]. 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 models are learnt for the textures from training images; and, classification of novel images. The 92 images for each texture are partitioned into two 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 images per texture defines a model for that texture as follows: the image is mapped (vector quantized) to a texton distribution (histogram). Thus, each texture is represented by a set of 46 histograms. An image from the test set is then classified by choosing the closest model histogram, and hence the corresponding texture class. The distance function used to define closest is the χ^2 significance test [16].

In all three experiments we follow both [2] and [13], and learn the texton dictionary from 20 textures (using the procedure outlined before in subsection 2.3). The particular textures used are specified in figure 7 of [13].

In the first experiment, 20 novel textures are chosen (see figure 5 in [2] 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 [13] are chosen. Here, a total of $40 \times 46 = 1840$ novel images are classified. Finally, in the third experiment, all the textures in the Columbia-Utrecht database are classified using the same procedure. The results for all three experiments are presented in table 1.

Table 1. Comparison of the classification rates for varying number of texture classes for each of the four filter sets. In all cases, 46 models are used for each texture class.

	# of texture classes		
filters	20	40	61
S	96.30%	95.16%	94.54%
LM	96.08%	93.75%	93.44%
MR4	94.13%	92.06%	90.76%
MR8	97.50%	96.30%	96.07%

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 clustering in a lower dimensional space is 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 this classification method is that as the number of texture classes increases there is only a small decrease in the classification rate.

3 Reducing the Number of Models

In this section, our objective is to reduce the number of models required for each texture class. In the previous section, the number of models was the same as the number of training images (and in effect [13] used 20 models/images for every texture). Here we want to reduce the number of models to that appropriate for each texture, independent of the number of training images. As is demonstrated in figure 9 some textures may require far fewer models than others. We will investigate two approaches for reducing the number of models. The first is selection of the histograms to determine the "natural" models for a particular texture. The second is pose normalization.

The classification experiments in this section are slightly modified so as to maximise the total number of images classified. If only M models per texture are used for training, then the rest of the $46 - M$ training images are added to the test set so that they too may be classified. Thus, for example when classifying 61 textures, if only $M = 10$ models are used then a total of 82 images per texture are classified giving a total of $82 \times 61 = 5002$ images. The texton dictionary used is the same as in the previous section and has 200 textons.

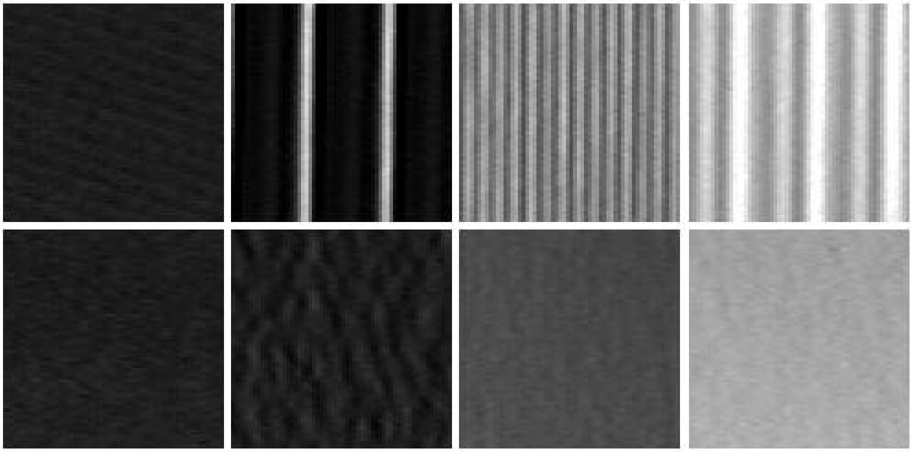

Fig. 9. Models per texture: the top row shows four images of the same texture, ribbed paper, photographed under different viewing and lighting conditions. The images look very different. The bottom row shows images of rough paper taken under the same conditions as the images in the first row. These images 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.

3.1 Selection to Determine Texture Models

We would expect that the number of different models that a texture requires 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.

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. Here, we investigate two schemes for model reduction, both of which take into account the inter and intra class image variation.

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 the $\mathbb{R}^n$ space. Therefore, given a distance function between two points, in our case χ^2, the *K-Medoid* algorithm may be used to cluster the histograms of *all* the training images (i.e. all the training images taken from all the texture classes) into representative centres (models). *K-Medoid* is a standard clustering algorithm [10] 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 *K-means*. In fact, *K-Means*can only be applied to each texture class locally. It can not be applied here as it merges data points and thus the resultant cluster centres can not be identified distinctly with individual textures. Table 2a 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.

Table 2. Classification rates for each of the four filter sets when the models are automatically selected by (a) the *K-Medoid* algorithm, and (b) the *Greedy* algorithm. In all cases there are 20 texture classes.

	Average # of models per texture		
filters	3	6	9
S	74.66%	86.40%	90.12%
LM	74.16%	86.69%	90.36%
MR4	69.94%	80.93%	85.36%
MR8	79.10%	89.59%	93.49%

(a)

	Average # of models per texture		
filters	3	6	9
S	88.37%	97.21%	98.01%
LM	86.69%	95.99%	97.83%
MR4	85.00%	93.66%	96.39%
MR8	90.67%	98.14%	98.61%

(b)

The classification rate with 9 *K-Medoid* selected models per texture is almost as good as using all 46 models (see column 1 in table 1). It should be noted here, that due to the increase in the size of the test set, many more novel images have to be classified correctly to achieve the same classification rate as in the previous section. Nevertheless, there is still significant room for improvement.

Greedy algorithm: An alternative to the *K-Medoid* clustering algorithm is a greedy algorithm designed to maximise the classification accuracy while minimising the number of models used. The algorithm is initialized by setting the number of models equal to the number of training images. 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. Figure 10, and table 2, show the resultant classification accuracy versus number of models for the four filter banks when classifying $20, 40$ and 61 textures. Figure 11 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.

A very respectable classification rate of over 97% correct is achieved for an average of 9 models per texture, even when all 61 texture classes are included. Furthermore, for most of the images that were misclassified, the correct texture class was ranked within the 5 most probable texture classes. It can therefore be hoped that incorporating extra information in the form of second order statistics or the co-occurrence of textons will lead to almost all the images being classified correctly (see future work).

These results compare very favourably with those reported in [2] and [13]. Cula and Dana obtain a classification rate of 96% while effectively using 11 models per texture. Using 5 models per texture, they achieve a classification

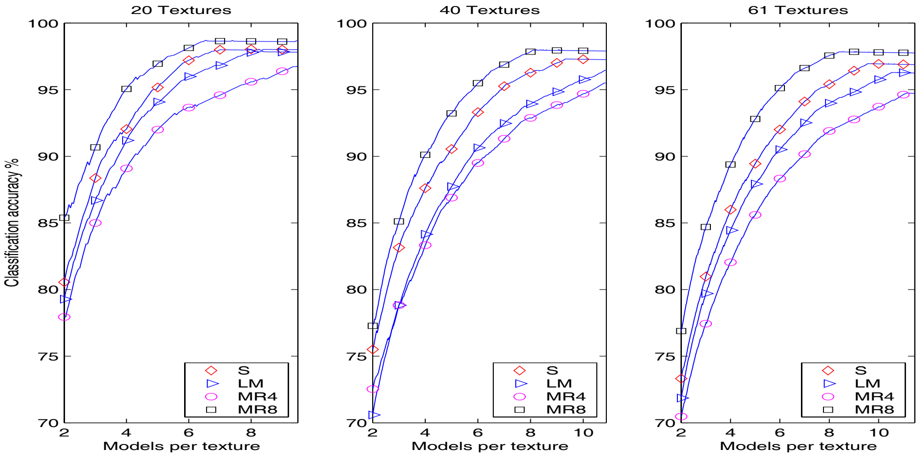

Fig. 10. Classification rates for models selected by the *Greedy* algorithm for 20, 40 and 61 textures.

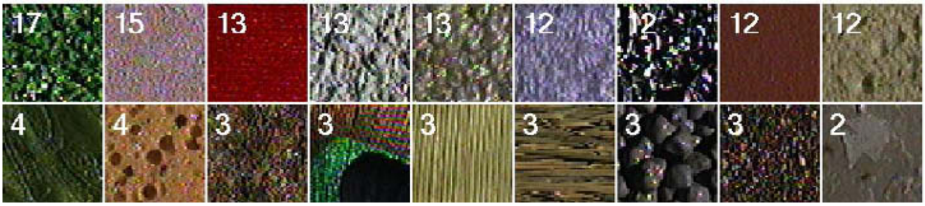

Fig. 11. Models selected by the *Greedy* algorithm while classifying all 61 textures: The top row shows the 9 texture classes, and the corresponding number of models, that were assigned the most number of models by the *Greedy* algorithm while the bottom row shows the 9 classes that were assigned the least number of models. Moving from left to right, the textures and the number of models assigned to it are: Artifical 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).

rate of roughly 87% (see figure 6 and table 2 in [2]). In contrast, again for 20 textures, the MR8 filter bank achieves a classification rate of 96% using 5 models per texture and achieves 98.4% using, on average, 10.4 models per texture. The main reason why the *Greedy* algorithm uses a fewer number of models for an equivalent classification rate is because the models that Cula and Dana learn are general models and not geared specifically to 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 and 87% for single images using an algorithm which effectively needs 20 models per texture. The MR8 filter bank achieves 95.6% accuracy on the same textures using only 6.05 models per texture, and furthermore achieves 98.01% accuracy using, on average, 9.07 models per texture.

3.2 Pose Normalization

In [17] it was demonstrated that, provided a texture has sufficient directional variation, it can be pose normalized by maximizing weak isotropy of the second moment gradient matrix (a method originally suggested in [14]). The method is applicable in the absence of solid texture effects. Here we investigate if this normalization can be used to at least reduce the effects of changing viewpoint,

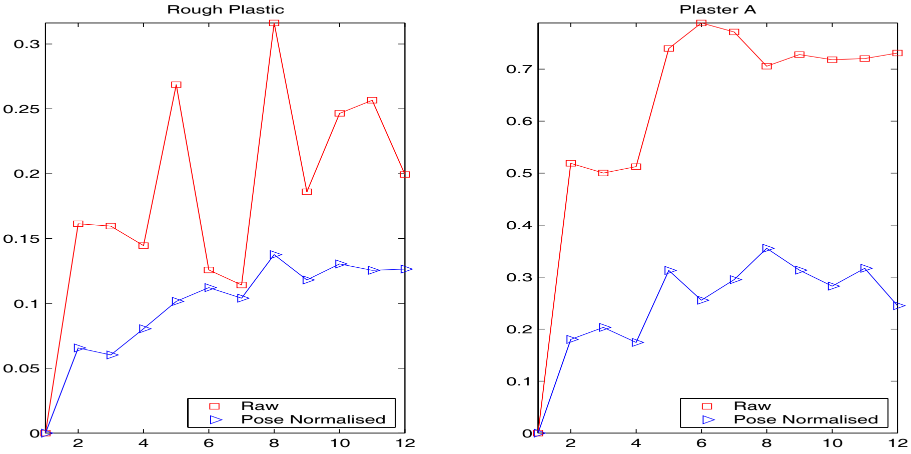

Fig. 12. The effect of pose normalization on a set of 12 images for two textures: "Rough Plastic" and "Plaster A". The 12 images have been sorted according to increasing viewing angle and this is represented on the X axis. The Y axis is the χ^2 distance between the model image and the given image. The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

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 normalization is successful, then for moderate changes in the viewing angle, two such "pose normalized" images of the same texture should differ from each other by only a similarity transformation. If there are no major changes in scale, 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 12 shows results for two textures - "Plaster A" and "Rough Plastic". Twelve images of each texture are selected to have similar photometric appearance, 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 normalization. As can be seen, the χ^2 distance is reduced for the pose normalized images. This in turn translates to better classification as well. On experiments on 4 textures, using the same 12 image set and 1 model per texture, the classification rate increased from 81.81% before pose normalization to 93.18% afterwards.

3.3 Generalization and Scale

In this subsection, we investigate how various factors affect our algorithm's classification rate. 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.

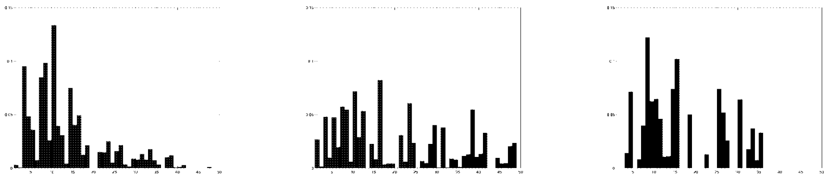
(a) (b) (c)

Fig. 13. Scaling the data results in new models: The histogram of texton labelings 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 texton vocabulary is sufficient, and it is the model that must be extended.

Initially, 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. Also, 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 rate of 98.3% when classifying all 61 textures using, on average, 8 models per texture.

To see if this choice of training images is important for our algorithm, the classification experiment was repeated using 46 training images that were chosen randomly from the 92 available. The average classification accuracy, computed over 10000 trial runs, was 98.16% and had stabilized to that value within 84 runs.

We also evaluated the "expressive power" of our texton dictionary by performing the experiment having compiled the textons from only 31 randomly chosen texture classes, giving a total of 310 textons. The classification rate decreased only slightly to 98.19% signifying that our textons are sufficiently versatile.

The number of textons in the dictionary can further be reduced by merging textons which lie very close to each other. We pruned down the size of the dictionary from 310 to 100 textons by selecting 80 of the most distinct textons (i.e. those textons that didn't have any other textons lying close by) and ran K-*Means*, with $K = 20$, on the rest. This procedure entailed another slight decrease in the classification accuracy to 97.38%.

Thus, while classifying 61 textures, the best classification rate achieved was 98.3% obtained when all 610 textons were used and the worst rate was 97.38% when only 100 textons wree used. We can therefore conclude that our algorithm is robust to the choice of training image set and texton vocabulary with the classification rate not being affected much by changes in these parameters.

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.

4 Conclusions and Future Work

We have demonstrated that with a handful of models per texture, classification rates superior to [2,13] can be achieved. In particular, only a single image is required for classification, not twenty with registration, and there is no need to supply the imaging conditions, either implicitly or explicitly. By developing the MR set, we are now in a position where we will be able to compare the "texton clustering and distribution" method of discriminating textures with the Bayesian approach proposed by Konishi and Yuille [12]. Their method stores the joint PDF of filter responses (in suitably sized bins), and is completely infeasible if the dimension of the filter space is large – [12] use a six dimensional space. For example, it would not be possible to use this approach with the 48 dimensional filter space used by [13]. Konishi and Yuille make strong use of colour, but achieve impressive results in classifying outdoor scenes into a small number of texture classes.

We also plan to investigate how the co-occurrence of textons (as in [18]) can lead to further improvements. Traditional rotation invariant filter banks are not very successful at measuring anisotropic features. However, the MR filter banks can detect such features accurately. Furthermore, recording the relative orientation at which the features occur will allow additional discriminability while preserving rotational invariance.

Finally, one other interesting point to investigate is whether the dimensionality of the MR filter response space can be reduced still further by not only taking the maximum response over orientations but also over all scales as well.

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

References

1. M. J. Chantler, G. McGunnigle, and J. Wu. Surface rotation invariant texture classification using photometric stereo and surface magnitude spectra. In *Proc. BMVC.*, pages 486–495, 2000.
2. O. G. Cula and K. J. Dana. Compact representation of bidirectional texture functions. In *Proc. CVPR*, 2001.

3. O. G. Cula and K. J. Dana. Recognition methods for 3d textured surfaces. Proceedings of the SPIE, San Jose, Jan 2001.
4. K. Dana and S. Nayar. Histogram model for 3d textures. In *Proc. CVPR*, pages 618–624, 1998.
5. K. Dana and S. Nayar. Correlation model for 3D texture. In *Proc. ICCV*, pages 1061–1067, 1999.
6. K. J. Dana, B. van Ginneken, S. K. Nayar, and J. J. Koenderink. Reflectance and texture of real world surfaces. *ACM Trans. Graphics*, 18,1:1–34, 1999.
7. R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* Wiley, 1973.
8. C. Fowlkes, D. Martin, X. Ren, and J. Malik. Detecting and localizing boundaries in natural images. Technical report, University of California at Berkeley, 2002.
9. G. M. Haley and B. S. Manjunath. Rotation-invariant texture classification using a complete space-frequency model. *IEEE PAMI*, 8(2):255–269, 1999.
10. L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data.* John Wiley & Sons, 1990.
11. S. Konishi, A. Yuille, J. Coughlan, and S. Zhu. Fundamental bounds on edge detection: An information theoretic evaluation of different edge cues. In *Proc. CVPR*, pages 573–579, 1999.
12. S. Konishi and A. L. Yuille. Statistical cues for domain specific image segmentation with performance analysis. In *Proc. CVPR*, 2000.
13. T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *IJCV*, Dec 1999.
14. T. Lindeberg and J. Gårding. Shape-adapted smoothing in estimation of 3-d depth cues from affine distortions of local 2-d brightness structure. In *Proc. ECCV*, pages 389–400, May 1994.
15. X. Liu, Y. Yu, and H. Shum. Synthesizing bidirectional texture functions for real-world surfaces. In *Proc. ACM SIGGRAPH*, pages 117–126, 2001.
16. W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C.* Cambridge University Press, 1988.
17. F. Schaffalitzky and A. Zisserman. Viewpoint invariant texture matching and wide baseline stereo. In *Proc. ICCV*, Jul 2001.
18. C. Schmid. Constructing models for content-based image retrieval. In *Proc. CVPR*, 2001.
19. A. Zalesny and L. Van Gool. A compact model for viewpoint dependent texture synthesis. volume 2018 of *Lecture Notes in Computer Science*, pages 124–143. Springer, July 2000.
20. S.C. Zhu, Y. Wu, and D. Mumford. Filters, random-fields and maximum-entropy (FRAME): Towards a unified theory for texture modeling. *IJCV*, 27(2):107–126, March 1998.