Texture Classification: Are Filter Banks Necessary?

Manik Varma
Robotics Research Group
Dept. of Engineering Science
University of Oxford
Oxford, UK OX1 3PJ
manik@robots.ox.ac.uk

Andrew Zisserman
Robotics Research Group
Dept. of Engineering Science
University of Oxford
Oxford, UK OX1 3PJ
az@robots.ox.ac.uk

Abstract

We question the role that large scale filter banks have traditionally played in texture classification. It is demonstrated that textures can be classified using the joint distribution of intensity values over extremely compact neighbourhoods (starting from as small as 3×3 pixels square), and that this outperforms classification using filter banks with large support.

We develop a novel texton based representation which is suited to modelling this joint neighbourhood distribution for MRFs. The representation is learnt from training images, and then used to classify novel images (with unknown viewpoint and lighting) into texture classes.

The power of the method is demonstrated by classifying over 2800 images of all 61 textures present in the Columbia-Utrecht database. The classification performance surpasses that of recent state-of-the-art filter bank based classifiers such as Leung & Malik [IJCV 01], Cula & Dana [CVPR 01], and Varma & Zisserman [ECCV 02].

1 Introduction

Texture research is generally divided into four canonical problem areas [7]: (1) synthesis; (2) classification; (3) segmentation; and (4) shape from texture. Significant progress was made during the 1990s on the first three areas (with shape from texture receiving comparatively less attention). The success in these areas was largely due to learning a fuller statistical representation of filter bank responses [1, 2, 10, 11, 13, 17]. It was fuller in three respects: firstly, the filter response *distribution* was learnt (as opposed to recording just the low order moments of the distribution); secondly, the *joint* distribution, or co-occurrence, of filter responses was learnt (as opposed to independent distributions for each filter); and thirdly, simply more filters were used than before – typically between ten and fifty filters or wavelets – to measure texture features at a set of scales and orientations.

These filter response distributions were learnt from training images and represented by clusters [2, 11, 15, 17], or histograms [9, 10, 19]. The distributions could then be used for classification, segmentation or synthesis. For instance, classification could be achieved by comparing the distribution of a novel texture image to the model distributions learnt from the texture classes. Similarly, synthesis could be achieved by constructing a texture having the same distribution as the target texture.

However, the supremacy of filter bank based methods was brought into question, in the case of texture synthesis, by the approach of Efros and Leung [6]. They demonstrated that superior synthesis results could be obtained using local pixel neighbourhoods directly, without resorting to large scale filter banks. In a related development, Zalesny and Van Gool [18] also eschewed filter banks in favour of a Markov Random Field (MRF) model.

Both these works put MRFs firmly back on the map as far as texture synthesis was concerned. Efros and Leung gave a computational method for generating a texture with similar MRF statistics to the original sample, but without explicitly learning or even representing these distributions. Zalesny and Van Gool, using a subset of all available cliques present in a neighbourhood, showed

that it was possible to learn and sample from a parametric MRF model given enough computational power.

In this paper, we show that the second of the canonical problems, texture classification, can also be tackled effectively by employing only local neighbourhood distributions, and without the use of large filter banks. In particular, we focus on the texture classification algorithm of Varma and Zisserman [17] which, to the best of our knowledge, currently gives the most accurate classification results on the Columbia-Utrecht (CUReT) database [3]. It correctly classifies over 96% of a test set of 2806 images taken from all 61 texture classes with unknown pose and illumination. We demonstrate that if the responses of the large scale filter bank used in this algorithm (with support up to 50×50 pixels square) are replaced by a feature space of pixel intensities determined over a small neighbourhood (e.g. 3×3), then similar, or even superior, classification results are obtained. This is a remarkable result, and we discuss why it holds given the MRF nature of textures.

The outline of the rest of the paper is as follows: in section 2 we review the filter bank based Varma and Zisserman (VZ) texture classifier. Next, in section 3, we develop an algorithm which classifies on the basis of local distributions instead and present comparisons with the VZ classifier. The new algorithm is referred to as the MRF classifier and in section 4 we discuss why it works so well.

2 A review of the VZ classifier

The classification problem being tackled is the following: given an image consisting of a single texture obtained under unknown illumination and viewpoint, classify it into one of a set of pre-learnt texture classes. Leung and Malik's influential paper [11] established much of the framework for this area - textons, χ^2 statistic, testing on the CUReT database. Later algorithms such as [2, 17] have built on this paper and extended it to classify single images without compromising accuracy.

In this section, we first describe the CUReT database and its level of difficulty for single image classification, and then overview the VZ classifier and its performance.

The CUReT database: There are 61 texture classes present in the database and each has been imaged under 205 viewing and illumination conditions. The effects of specularities, inter-reflections, shadowing and other surface normal variations are plainly evident and this makes the database far more challenging for a classifier than the often used Brodatz collection. The limitations of the database are a lack of significant scale change and limited in-plane rotation. Figure 1 shows how demanding a task single image classification can be for CUReT samples.

In this paper, we replicate the setup of [17] and include all 61 classes present in the database. From each class, 92 images are selected (with only the most extreme viewpoints being excluded) and partitioned into two disjoint sets of 46 images each. Images in the first (training) set are used for model learning while classification accuracy is assessed on the second (test) set. Thus, there are $61 \times 46 = 2806$ training images and 2806 test images.

The VZ classifier: The classifier is divided into two stages: a learning stage where statistical distribution models of texture classes are learnt from training examples, and a classification stage where novel images are classified by comparing their distributions to the learnt models.

In the learning stage, training images are convolved with a chosen filter bank to generate filter responses. These filter responses are then aggregated over images from a texture class and clustered. The resultant cluster centres form a dictionary of exemplar filter responses which are known as textons. Given a texton dictionary, a

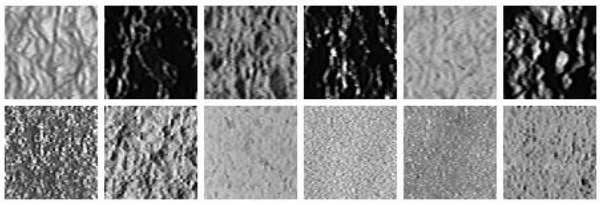

Figure 1: Single image classification on the CUReT database is a demanding task. In the top row, the images differ markedly from each other (due to variation in illumination and pose) even though they all belong to the same texture class. This illustrates large intra class variation. In the bottom row, several of the images look similar and yet belong to different texture classes. This illustrates that the database also has small inter class variation.

model is learnt for a particular training image by labelling each of the image pixels with the texton that lies closest to it in filter response space. The model is the normalised frequency histogram of pixel texton labellings, i.e. an n-vector of texton probabilities for the image, where n is the number of textons. Each texture class is represented by a number of models corresponding to training images of that class.

In the classification stage, the set of learnt models is used to classify a novel (test) image into one of the 61 textures classes. This proceeds as follows: the filter responses of the test image are generated and the pixels labelled with textons from the texton dictionary. Next, the normalised frequency histogram of texton labellings is computed to define an n-vector for the image. A nearest neighbour classifier is then used to assign the texture class of the nearest model to the test image, where the distance between two normalised frequency histograms is measured using χ^2 [14].

In previous work [17], the performance of four filter banks was contrasted (including the filter bank used by Leung and Malik (LM) [11] and Cula and Dana [2] as well as the one used by Schmid (S) [16]) and it was demonstrated that the rotationally invariant, multi-scale, Maximum Response MR8 filter bank (described below) yields better results than all the other three. Hence, in this paper, we present results and make comparisons with MR8.

Filter bank: The MR8 filter bank consists of 38 filters but only 8 filter responses. The filters include a Gaussian and a Laplacian of a Gaussian (LOG) filter both at scale $\sigma = 10$, an edge (first derivative) filter at 6 orientations and 3 scales and a bar (second derivative) filter also at 6 orientations and the same 3 scales (σ_x,σ_y)=$\{$(1,3), (2,6), (4,12)$\}$. The response of the isotropic filters (Gaussian and LOG) are used directly, but the responses of the oriented filters (bar and edge) are "collapsed" at each scale by using only the maximum filter responses across all orientations - thereby giving 8 filter responses in total and ensuring that each filter in the filter bank is rotationally invariant. The MR4 filter bank only employs the $(\sigma_x, \sigma_y) = (4, 12)$ scale. Further details of the filter bank, as well as pre and post image processing steps, are given in [17].

Greedy algorithm: The question remains of how many models are required to characterise each texture class. One possibility is to use all the 46 training images per class. Another is a greedy algorithm designed to iteratively maximise the classification accuracy while minimising the number of models used. This proceeds as follows, an initial list of models is drawn from the training set. Next, at each iteration step, one model is discarded from the list. This model is chosen to be the one whose exclusion least affects the classification performance. Iterations are repeated until no more models are left. The classification accuracy can be measured by partitioning the training set into a learning set and a validation set. The greedy algorithm selects models from the learning set while classification accuracy during iterations is assessed on the validation set.

Implementation details and results: To learn the texton dictionary, filter responses of 13 randomly selected images per texture class (taken from the set of training images) are aggregated and clustered via the *K-Means* algorithm [4]. $K = 10$ textons are learnt from each of the 61 texture classes present in the CUReT database resulting in a dictionary comprising $61 \times 10 = 610$ textons.

Under this setup, the VZ classifier achieves an accuracy rate of 96.93% while classifying all 2806 test images into 61 classes using 46 models per texture. The Greedy algorithm reduces the number of models to, on average, just 8 per texture down from the original 46. In [17], results are also reported for the case where the images excluded from the training set by the Greedy algorithm are added to the test set and classified. A classification accuracy of 98.3% is achieved. However, in this paper we will keep the training and test sets distinct and, in the next section, only present comparisons with classification carried out solely on the test set.

The classification results for 61 textures using 610 textons for the other three filter banks investigated in [17] are: Maximum Response 4 (MR4) = 91.69%, Leung and Malik (LM) = 94.65% and Schmid (S) = 95.22%. For MR8 the best classification results of 97.43% are obtained when a dictionary of 2440 textons is used, with 40 textons being learnt per texture class.

3 The MRF classifier

We now develop a classifier which uses the complete local neighbourhood of pixel values, rather than filtered responses. It is demonstrated that small neighbourhoods of size 3×3 through to 7×7 achieve superior classification performance to multi-scale filter banks with large support.

The new algorithm is based on the VZ classifier of section 2, but at the filtering stage, instead of using the MR8 filter bank to generate filter responses at a point, the raw pixel intensities of an $N \times N$ square neighbourhood around that point are taken and row reordered to form a vector in an N^2 dimensional feature space. All pre and post processing steps are retained and no other changes are made to the classifier. The results for this classifier using 610 textons are given in table 1(a). It is remarkable to note that classification results of over 95% are achieved using neighbourhoods as small as 3×3. In fact, the classification result for the 3×3 neighbourhood is actually better than the results obtained by using the MR4, LM or S filter banks in the VZ classifier. This is strong evidence that there is sufficient information in the joint distribution of the nine intensity values (the central pixel and its eight neighbours) to discriminate between the texture classes. We will refer to this classifier as the Joint classifier.

N	All points (a)	All points but central (b)	MRF with 90 bins (c)
3	95.33% (9.6)	94.90% (9.3)	95.87% (9.4)
5	95.62% (8.4)	95.97% (8.6)	97.22% (8.1)
7	96.19% (8.4)	96.08% (8.2)	97.47% (7.9)

Table 1: Comparison of classification rates of all 61 textures in the CUReT database for different $N \times N$ neighbourhood sizes. (a) all points in the neighbourhood are used to form vectors in an N^2 feature space. (b) all points but the central point are used (i.e. an $N^2 - 1$ space). (c) the MRF classifier where 90 bins are used to represent the conditional central pixel PDF. The bracketed values report the number of models per texture class as determined by the Greedy algorithm. A dictionary of 610 textons is used throughout. Notice that comparable, and even superior, performances to MR8's (of 96.93% using 610 textons, and 97.43% using 2440 textons) are achieved.

To see why the Joint classifier achieves such good results, we investigate to what extent textures may be considered realisations of an MRF, as measured by classification. Formally, for an MRF

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

where $\mathbf{x}_c$ is a site in the 2D integer lattice on which the image I has been defined and $\mathcal{N}(\mathbf{x}_c)$ is the neighbourhood of that site. In our case, we have defined $\mathcal{N}$ to be the $N \times N$ square neighbourhood (excluding the central pixel). Thus, although the value of the central pixel is significant, its distribution is conditioned on its neighbours alone. To test how tightly this distribution is conditioned, the classifier is retrained on feature vectors drawn only from the set of $\mathcal{N}$: i.e. the set of $N \times N$ neighbourhoods with the central pixel left out. For example, in the case of a 3×3 neighbourhood, only the 8 neighbours of every central pixel are used to form feature vectors and textons. This is referred to as the Neighbourhood classifier.

As shown in table 1(b), there is almost no significant variation in classification performance compared to using all pixels in the $N \times N$ region. Classification rates for $N = 5$ are slightly better when the central pixel is left out and marginally poorer for the cases of $N = 3$ and $N = 7$. Thus, the joint distribution of the neighbours is largely sufficient for classification. This supports the validity of the MRF model for the textures in this database.

We now go to the other extreme, and instead of ignoring the central pixel, we explicitly model $p(I(\mathbf{x}_c)|I(\mathbf{x}), \mathbf{x} \in \mathcal{N}(\mathbf{x}_c))$, i.e. the distribution of the central pixels conditioned on their neighbours. Up to this point, we have used textons to represent the joint PDF of the central pixels and their neighbourhoods. This is now modified slightly to represent the PDF of the central pixels explicitly conditioned on their neighbourhoods.

To learn the conditional PDF representing the MRF model for a given training image, we first represent the neighbours' PDF by textons as above – i.e. all pixels but the central are used to form feature vectors in an $N^2 - 1$ dimensional space which are then labelled using the same dictionary of 610 textons. Then for each of the n textons in turn, a one dimensional distribution of the central pixels' intensity is learnt and represented by an m bin histogram. Thus the representation of the joint PDF is now an $n \times m$ matrix. Each row is the PDF of the central

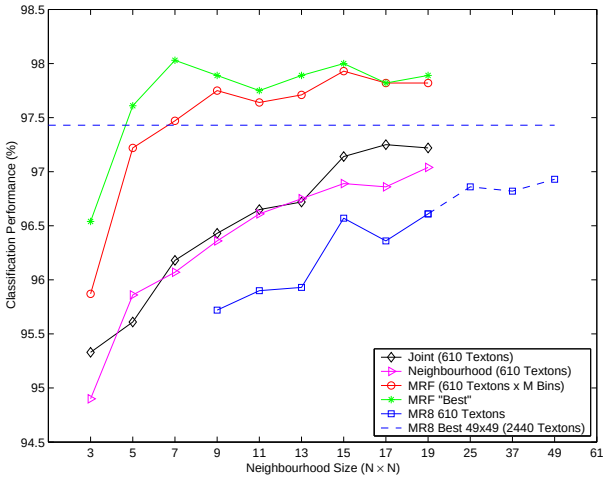

Figure 2: Classification results as a function of neighbourhood size. The performance of the MR8 filter bank is always worse than any other classifier at the same neighbourhood size. The MRF "Best" curve shows results obtained for the best combination of texton dictionary and number of bins for a particular neighbourhood size. For neighbourhoods up to 11×11, dictionaries of up to 3050 textons and up to 200 bins are tried. For 13×13 and larger neighbourhoods, the maximum size of the texton dictionary is restricted to 1220 because of computational expense. The best result achieved by the MRF classifiers is 98.03% using a 7×7 neighbourhood with 2440 textons and 90 bins. The best result for MR8 is 97.43% for a 49×49 neighbourhood and 2440 textons.

pixel conditioned on a particular neighbourhood intensity configuration as represented by a specific texton. This is somewhat similar to the co-occurrence representation of [16] for filter banks.

Using this texton representation, a novel image is classified by comparing its MRF distribution to the model MRF distributions (learnt from training images) using χ^2 over all elements of the $n \times m$ matrix. We will refer to this as the MRF classifier. Table 1(c) compares the classification performance to the previous classifiers, when a resolution of 90 bins is used for the central pixels' histograms. In all cases, there are 46 models per texture and a dictionary of 610 textons is used. As can be seen, at each neighbourhood size (3×3, 5×5 and 7×7), the performance of the MRF classifier is superior to the other classifiers. In fact, the result for the 7×7 case is better than the best performance achieved for MR8 (97.43% using 2440 textons).

Up to now, modelling the full conditional PDF was considered infeasible, and either parametrised Gibbs potentials [8, 18] or at best a combination of marginals [19] were used. The texton representation developed here is different from traditional MRF models which learn potential functions and then use the Hammersley-Clifford theorem to calculate the joint probability $p(I)$ [12]. We, on the other hand, do not explicitly learn any potential functions and model the MRF distribution by the learnt conditional probabilities $p(I(\mathbf{x}_c)|I(\mathcal{N}(\mathbf{x}_c))$.

It is worth reflecting on why textons are able to adequately represent the PDF of texture neighbourhoods in this manner. Since a homogeneous texture is by nature repetitive, albeit with small statistical variations, it is expected that across the image the co-occurrence of neighbours should form clusters. Thus a cluster based representation of the PDF using a few textons should suffice even in very high dimensional spaces, since most of the space is empty and need not be modelled.

We turn now to the question of whether filter banks are providing beneficial information for classification, for example perhaps by increasing the signal to noise ratio, or by detecting features such as bars, edges and spots. We compare the performance of the VZ classifier using the MR8 filter bank to that of the Joint classifier, Neighbourhood classifier, and MRF classifier as the size of the neighbourhood is varied. The MR8 filter bank is scaled down so that the support of the largest scale filters is the same as the neighbourhood size. Figure 2 plots the results. It is interesting to note that for any given size of the neighbourhood, the MR8 classifier performs worse than the Joint classifier and even the Neighbourhood classifier. It would thus appear that using all the information present in an image patch is more beneficial for classification than relying on pre-selected filter banks. A classifier which is able to learn from all the pixel values is superior. The MRF classifier achieves 98.03% using a 7×7 neighbourhood with 2440 textons and 90 bins. The Greedy Algorithm reduces the number of models used to 7.27 on average. This means that only 55 images are classified incorrectly out of 2806. In contrast, the best performance of the MR8 classifier is 97.43% for a 49×49 neighbourhood and 2440 textons.

4 Why does the MRF classifier work?

It was demonstrated in the previous section that a classification scheme based on MRF local neighbourhood distributions can achieve very high classification rates and can outperform filter bank based methods.

What has been shown is that textures with global structures far larger than the local neighbourhoods used can be classified (discriminated) by a distribution of local measurements. The explanation for this is illustrated in figure 3 where three images are selected from two texture classes (Limestone and Ribbed Paper) of the CUReT dataset, and scatter plots of their grey level co-occurrence matrix shown for the displacement vector (2,2). Notice how the distributions of the two images of Ribbed Paper can easily be associated with each other and distinguished from the distribution of the Limestone image. Thus 3×3 neighbourhood distributions can contain sufficient information for successful discrimination.

This raises two questions: first, what classes of (texture) signal can be discriminated on the basis of measured local distributions? And, second, since the textures of the CUReT dataset evidently are in the class which can be distinguished, what neighbourhood size is required for this dataset?

The MRF "Best" curve (figure 2) can help answer the second question. The classification accuracy first increases with increasing neighbourhood size, reaches a maximum for a 7×7 neighbourhood using 2440 textons and then goes down slightly for 9×9 and 11×11. This indicates that the optimal neighbourhood size for the CUReT dataset is around 7×7.

To tackle the question of what classes of signal can be distinguished by local measurements alone we consider some one dimensional examples. In the case of a polynomial of degree $2N - 1$, the Taylor series expansion immediately shows us that a $[-N, +N]$ neighbourhood contains enough information to *determine* the value of the central pixel. Similarly, the central point for periodic functions such as sines and cosines can be determined from a small neighbourhood. Of course, in general, synthesis requires much more information than classification and therefore it is expected that more complicated functions can still be distinguished just by looking at small neighbourhoods. For example, a square wave can be differentiated from a triangular wave by looking at the distribution of gradients (which can be determined from a neighbourhood of just two points). Similarly, sine waves of different frequencies can also be distinguished just by looking at two point neighbourhoods. In the extreme case, two arbitrarily similar discrete functions (no matter how complex) which differ only in the number of times they attain one particular value can be distinguished by looking at just the central pixel and noting how many times it attains that value.

There also exist entire classes of functions which can not be distinguished on the basis of local information alone. For instance, any two textures which have identical first order texton statistics but which differ in their

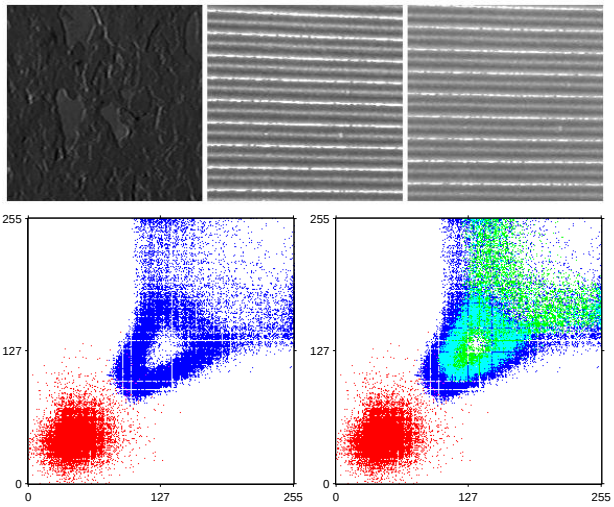

Figure 3: Information present in 3×3 neighbourhoods is sufficient to distinguish between textures. The top row shows three images drawn from two texture classes, Limestone and Ribbed Paper. The bottom row shows scatter plots of $I(\mathbf{x})$ against $I(\mathbf{x} + (2, 2))$ for (left) the Limestone and first Ribbed paper images, and (right) all three images. The Limestone and Ribbed Paper distributions can easily be distinguished (Limestone is the bottom left cluster in red and Ribbed Paper the top right rotated 'A' in blue), and hence the textures can be discriminated from this information alone.

higher order statistics will be indistinguishable. To take a simple example, consider texture classes generated by the repeated tiling of two textons (a circle and a square for instance) with sufficient spacing in between so that there is no overlap between textons in any given neighbourhood. Then, any two texture classes which differ in their tiling pattern but have identical frequencies of occurrence of the textons will not be distinguished on the basis of local information alone. Further research is needed to identify such classes of functions and to isolate them from those that can be distinguished by local distributions. However, the fact that we can achieve over 98% accuracy using 7×7 neighbourhoods indicates that the CUReT textures do not belong to such function classes.

5 Scale, rotation and synthesis

It could be argued that the lack of significant scale change in the CUReT images might be the reason that the MRF classifier outperforms MR8. To test this hypothesis, 4 texture classes (# 2, 11, 12 and 14) were selected for which additional zoomed in data is available (as # 29, 30, 31 and 32) and the images were combined. When classifying the original textures, both the MRF and MR8 classifiers achieve 100% accuracy. When the zoomed in images are added to just the test set, the accuracy rates drop to 93.48% and 81.25% respectively, showing that the MRF classifier is not being adversely affected by the scale variations. When the zoomed in images are added to both the test and training sets, the accuracy rates go back to 100% and 99.46% respectively. This test was repeated when the images were zoomed artificially by a factor of 2. The results followed the same pattern where both the MRF and MRF8 classifiers achieved 100% before addition, 65.22% and 62.77% after addition to the test set, and 99.73% for both after addition to the test and training sets. Again, the results show that the MRF classifier has the same ability to cope with scale changes as the multi-scale filter bank MR8.

We next turn to the issue of rotational invariance. In [17] it was demonstrated that using rotationally invariant filter banks (i.e. the MR8 and S sets) gave superior classification performance to that of LM, which is not rotationally invariant. It is natural to question whether the performance of the MRF classifier could be enhanced by using a rotationally invariant representation of the neighbourhood. To achieve rotation invariance, instead of working with an $N \times N$ square patch, the neighbourhood is redefined to be circular with a given radius. The local orientation is determined for each circular neighbourhood, and then, before forming feature vectors, each neighbourhood is rotated by the determined angle to achieve invariance. The Neighbourhood classifier using this setup with a dictionary of 610 textons gives a classification accuracy of 96.36% for a radius of 3 pixels (corresponding to a 7×7 patch) and 96.47% for a radius of 4 pixels (corresponding to a 9×9 patch). These values are better than those reported by the "Square" Neighbourhood classifier with equivalent support. However, for the MRF classifier the rates are only 97.07% and 97.25% respectively using 610 textons and 45 bins. These are less than those reported by the "Square" MRF classifier. Further investigation is required in this area.

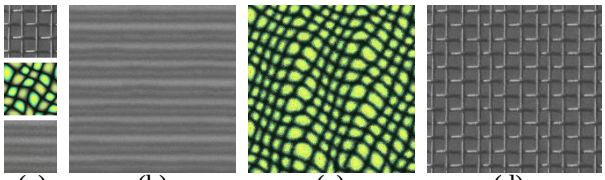

Figure 4: Synthesis: (a) Input texture blocks, (b) Ribbed Paper (CUReT) synthesised using a 7×7 neighbourhood and 100 textons (c) Efros and Leung [6] - 15×15, 800 textons and (d) D6 (Brodatz) - 11×11, 300 textons.

To conclude we demonstrate that our MRF representation may also be used for texture synthesis. The algorithm used is similar to [5, 6]. First, the MRF statistics of the input texture block are learnt using our representation. The parameters that can be varied are N, the size of the neighbourhood, and K the number of textons used to represent the neighbourhood distribution. The central pixel PDF is stored in 256 bins in this case. Next, to synthesise the texture, the input texture block is initially tiled as in [5] to the required dimensions. A new image is synthesised from this tiled image by taking every pixel, determining its neighbourhood (i.e. closest texton) and setting the value of the pixel to a value sampled from the learnt MRF distribution. This iteration is repeated until a desired synthesis is obtained. Results are shown in figure 4.

6 Discussion and conclusions

Filter banks have become ubiquitous in the texture classification literature over the last decade or so. The main reason for their popularity is biological plausibility and the hypothesis that many features at multiple orientations and scales need to be extracted accurately for successful classification. The work in this paper, following that of Efros and Leung [6], demonstrates that for certain tasks (synthesis, classification) filter banks are *not* necessary, but are sufficient (though their performance for both tasks is inferior).

Indeed filter banks have a number of disadvantages compared to smaller MRF neighbourhoods: first, the large support they require means that far fewer samples of a texture can be learnt from training images (there are many more 3×3 neighbourhoods than 50×50 in an 100×100 image). Second, the large support is also detrimental in texture segmentation, where boundaries are localised less precisely due to filter support straddling two regions; A third disadvantage is that the blurring (e.g. Gaussian smoothing) in many filters means that fine local detail can be lost. This is another reason why the MRF classifier achieves superior results to MR8.

The disadvantage of the MRF representation is the quadratic increase in the dimension of the feature space with the scale of the neighbourhood. This problem may be tackled by using a multi-scale representation – which filters traditionally provide – or an alternative method of selecting important long range interactions in the MRF, as is done by [18].

In conclusion, this paper has introduced a novel representation of the MRF distribution. It has also demonstrated that superior classification results can be obtained by using compact, local neighbourhoods and without the use of filter banks.

Acknowledgements

Financial support for this research was provided by a University of Oxford Graduate Scholarship in Engineering, an ORS award, and the EC project CogViSys.

References

[1] J. S. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. In *Proc. ACM SIGGRAPH*, 1997.

[2] O. G. Cula and K. J. Dana. Compact representation of bidirectional texture functions. In *Proc. CVPR*, pages 1041–1047, 2001.

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

[4] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, 1973.

[5] A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *Proc. ACM SIGGRAPH*, pages 341–346, 2001.

[6] A. Efros and T. Leung. Texture synthesis by non-parametric sampling. In *Proc. ICCV*, pages 1039–1046, 1999.

[7] D. A. Forsyth and J. Ponce. *Computer Vision: A modern approach*. Prentice Hall, 2002.

[8] S. Geman and D Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE PAMI*, 6(6):721–741, 1984.

[9] D. J. Heeger and J. R. Bergen. Pyramid-Based texture analysis/synthesis. In *Proc. ACM SIGGRAPH*, pages 229–238, 1995.

[10] S. Konishi and A. L. Yuille. Statistical cues for domain specific image segmentation with performance analysis. In *Proc. CVPR*, pages 125–132, 2000.

[11] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *IJCV*, 43(1):29–44, 2001.

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

[13] J. Portilla and E. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *IJCV*, 40(1):49–70, 2000.

[14] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.

[15] T. D. Rikert, M. J. Jones, and P. A. Viola. A cluster-based statistical model for object detection. In *Proc. ICCV*, 1999.

[16] C. Schmid. Constructing models for content-based image retrieval. In *Proc. CVPR*, volume 2, pages 39–45, 2001.

[17] M. Varma and A. Zisserman. Classifying images of materials: Achieving viewpoint and illumination independence. In *Proc. ECCV*, volume 3, pages 255–271. Springer-Verlag, 2002.

[18] A. Zalesny and L. Van Gool. A compact model for viewpoint dependent texture synthesis. In *Proc. ECCV*, LNCS 2018/5, 2000.

[19] 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.