COMPACT ROTATION-INVARIANT TEXTURE CLASSIFICATION

Paul Southam and Richard Harvey

School of Computing Sciences, University of East Anglia, Norwich, NR4 7TJ, UK.

ABSTRACT

This paper constructs a texture feature extractor based on a morphological scale-space. It produces features that are invariant to rotation of the texture. The features are used with a very simple *k*-nearest neighbour classifier and tested using the Outex methodology. The classifier has comparable performance to a number of benchmark classifiers but uses fewer features. The algorithm is quick to compute and provides insight into the underlying structure of texture.

1. INTRODUCTION

Texture is abundant in the natural world, so texture analysis is regarded as an important area in machine vision. This has lead to many texture segmentation, synthesis and classification methods. For classification, filter-bank based methods such as [6] have been shown to be successful and more recently, newer filter-bank based classifiers such as [4, 12] have built upon that success. However, the cost is often an increase in the number of features used in classification computations. Furthermore, since these filter-banks extract texture features at multiple scales and orientations, they are often sensitive to shifts in scale and rotation.

In this paper we introduce a compact rotation-invariant texture classifier based upon the *sieve* [1] which is a morphological scale-space operator that filters an input signal by removing intensity extrema at a specific scale. These removed extrema are known as *granules* and may be summed to re-create the original signal without loss. Because objects in images are often delineated by iso-intensity contours, sieves have been applied to image segmentation tasks in which these semantically meaningful objects are removed at specific (typically higher) scales. However, at smaller scales, the sieve can be used to remove image noise and, at increasing scale remove textural information as in Figure 1. By sieving a textured image at increasing scales and taking the difference between adjacent output images we can capture texture features at specific scales.

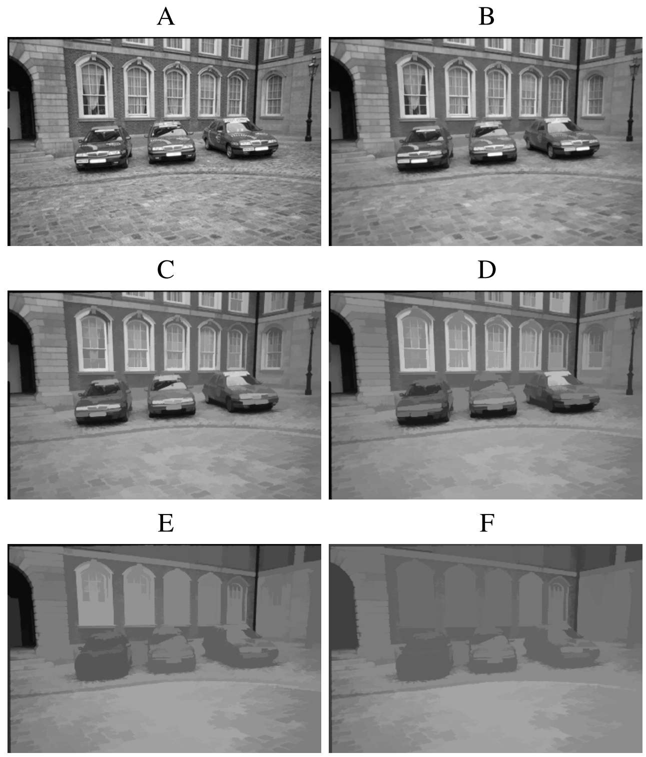

Fig. 1. An original image (A) sieved to scales 15(B), 90(C), 251(D), 2000(E) and 5000(F). Each image has fewer intensity extrema than its predecessor. A full decomposition may be summed to re-create the original thus the sieve is a *transform* of the original image.

2. METHODS

Texture classification is a frequent topic in the computer vision literature and capturing data to classify is often a time-consuming and tedious task. Therefore data are often taken from existing and well known texture databases such as: Brodatz [3]; VisTex [13]; MeasTex [10]; CUReT [5] and Outex [7] (see Table 1). Although Brodatz is the most commonly used, in this paper we opt not to use it for reasons stated in [7, 10]. Since the CUReT database is primarily designed for use with analysing and synthesizing texture on 3D objects and VisTex and MeasTex both have limited of classes and/or samples; we use Outex in our classification experiments.

0-7803-8554-3/04/$20.00 ©2004 IEEE.

	Brodatz	VisTex	MeasTex	Outex	CUReT
No. Classes	100–112*	19	4	29	61
No. Images per class	1–4*	1–20	4–25	1–47	205†
Evaluation Framework?	×	×	√	√	×
Defined test/training data?	×	×	√	√	×
Unique copy?	×	√	√	√	√

Table 1. Available texture databases (* Depending on implementation, † At differing angle and illuminant.)

Currently the Outex database consists of 319, grey-scale and colour textures spanning 29 different classes and a small selection of natural scenes. The images are organised into a number of *test suites* setup for, classification, segmentation and retrieval experiments. Some of these contain images from the Brodatz album. A number of these test suites are designed to test for illumination, scale, rotation and colour invarience. The great advantage of the Outex database is that each test suite has defined testing and training data. The importance of segregated testing and training data was illustrated in [11] and [9] where conflicting results might for example be explained by different testing and training data. Some Outex textures are shown in Figure 2.

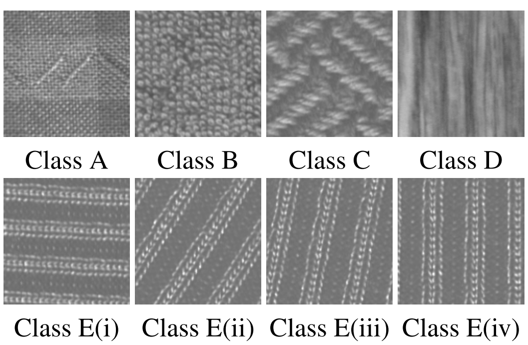

Fig. 2. Various Outex samples (top row) and rotated versions of the same texture class (bottom row)

The feature extraction process, based upon a M-sieve [2], is shown in Figure 3 and is as follows: each texture image was sieved to N different scales, $[S_1 \ldots S_N]$ where $log_{10}S_n$ are equispaced between 0 and $log_{10}P$, where $N = 11$ and $P = 251$ were chosen to remove all textural information from all images and give ten channels over which the texture might vary. Granule images, G_n, are then formed by subtracting each sieved image from one sieved to a previous scale, $G_n = S_n - S_{n-1}$; $n = [2 \cdots N]$. Thus the G_n are a transform of the texture which can be reconstructed through

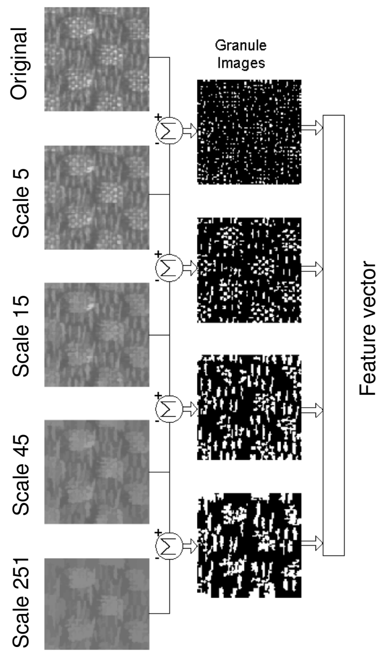

Fig. 3. The feature extraction process. The granule images are bipolar. Here we display only the positive pixels in white.

a simple summation. In this sense the proposed feature is not based upon a model of texture but on the properties of a digital image.

The magnitude of the granule images as a function of scale is an indicator of the distribution of the texture features by scale. Figure 4 shows the sample mean, standard deviation and skewness of the granule images for three different textures. Figure 4(i) shows that fine scale textures, such as class A, have a peak response at small scales; and coarser textures such as class B and C have peaks at higher scales. In these examples the standard deviation is correlated with the mean but this is not always the case. Figure 4(iii) shows the skew which measures the light/dark distribution as a function of scale.

The mean, standard deviation and skewness of the magnitude of granule images are used to form a feature vector 30 elements long. This feature vector is normalised by subtracting the mean and dividing by the standard deviation. Then, using principal component analysis (PCA), the data maybe reprojected onto an aligned axis with the further possibility to reduce feature dimension. During classification, the Euclidean distances are measured between each feature vector in the PCA space and a k nearest neighbour classifier is used to predict the test class.

The scales of extrema are functions of both amplitude and

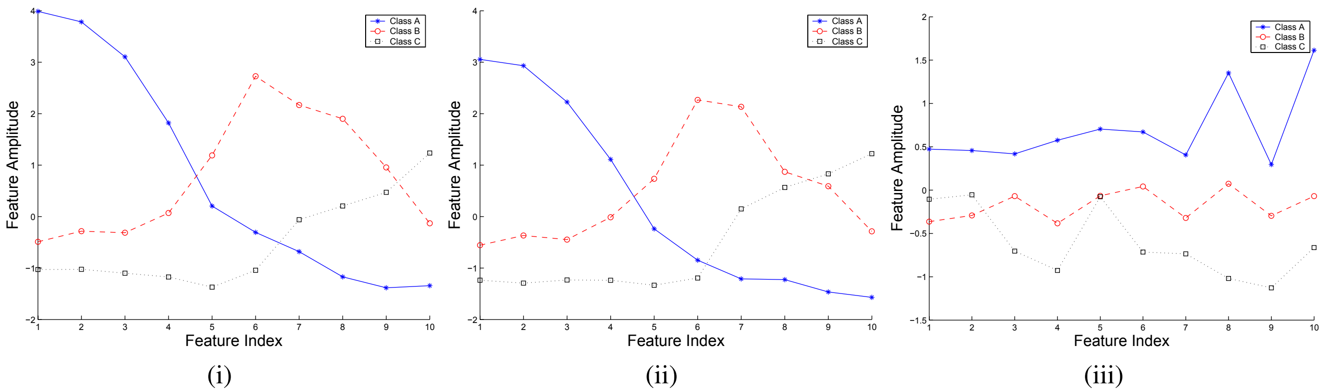

Fig. 4. The sample mean (i), the sample standard deviation (ii) and the sample skewness (iii) of three texture classes A,B and C shown in Figure 2.

area. Ignoring the finite size of pixels, simply rotating the image has no effect on either extrema amplitude or area, so its decomposition remains the same (see Figure 2). Therefore the features derived from rotated decompositions of the same image are identical.

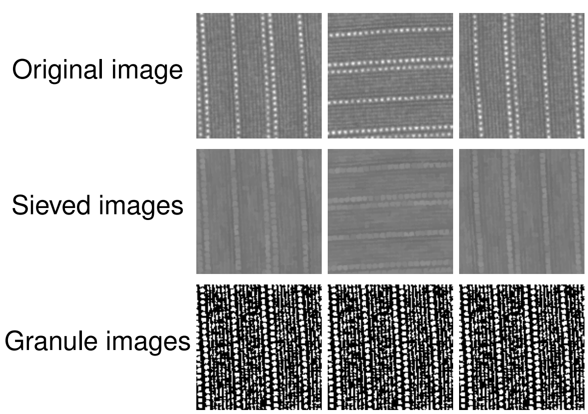

Fig. 5. A rotated Outex texture is sieved to a scale of 20. The re-orientated granule images are identical and so are the features derived from them.

3. RESULTS

The following test suites are used:

- **Outex_TC_00000**; **Outex_TC_00003** – each comprising 100 standard, 100 fold leave-out-half cross validation, classification experiments (different permutations of 240 testing and 240 training data) of 480 different images. There are 20 samples of 24 texture classes.
- **Outex_TC_00010** – one rotation-invariant classification test comprising 4320 images. There are 24 classes of 180 images at rotations of 00,05,10,15,30,45,60,75,90 degrees. 480 images used for training and 3840 for the one test set.

Table 2 shows the mean success rate using 30 features and PCA. Best results were achieved $k = 1$ which suggests that the Outex set would benefit from more data.

k	1	3	5	7	10
success rate	0.962	0.960	0.959	0.957	0.953
k	15	23	40	100	
success rate	0.944	0.920	0.910	0.931	

Table 2. Success rate $1-e$ where e is the mean error rate (computed over 100 trials) as a function of k

Table 3 show a comparison with Gabor wavelets (Gabor) [6], Local Binary Pattern (LBP) [8] and Gaussian Markov Random Fields (GMRF) taken from [7]. The sieve method has the second lowest performance but is the most consistent (lowest standard error). Under a t-test the sieve is indistinguishable from the GRMF (at all significance $\alpha = 0.1, 0.05, \cdots$) however the error distributions are far from Gaussian so we restrict ourselves to the observation that the error distributions overlap[1].

Using PCA to reduce feature dimensionality from 30 to 12 incurs little reduction in classification performance (typically $< 0.01\%$). Further reducing to 3 features results in a success rate of 0.92 with test suite **Outex_TC_00000**. Figure 6 shows the effect of reducing the number of features

[1]Unfortunately we do not have access to individual test results so we cannot use a more sensitive test such as McNemar's test.

	00000				00003			
	Sieve	Gabor	LBP	GMRF	Sieve	Gabor	LBP	GMRF
$\bar{x}$	0.962	0.995	0.995	0.961	0.965	0.996	0.995	0.964
max	0.992	1	1	0.992	0.992	1	1	0.992
min	0.933	0.983	0.983	0.925	0.929	0.983	0.973	0.933
σ	0.01	0.5	0.4	1.3	0.01	0.4	0.5	1.4
σ'	0.001	0.05	0.04	0.13	0.001	0.04	0.05	0.14

Table 3. $\bar{x}$ = 1 - e (success rate) , max and min success rate, standard deviation σ and standard error of the mean $\sigma' = \frac{\sigma}{\sqrt{N}}$ for the different Outex test suites.

for **Outex_TC_00000**.

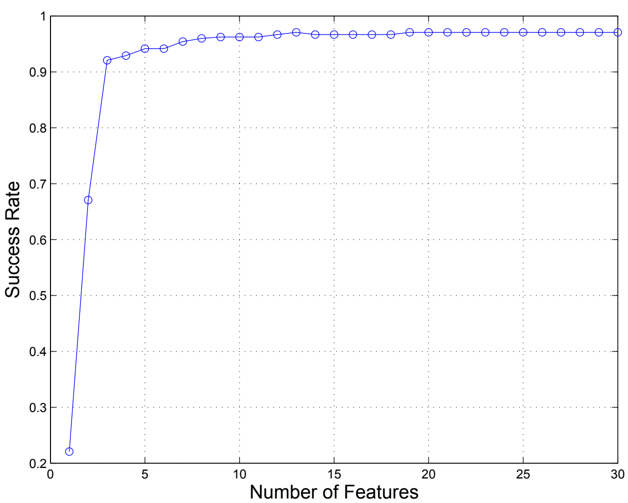

Fig. 6. Success rate against number of features

With the rotation invariant **Outex_TC_00010** test the sieve classification success rate is 0.948. Unfortunately other authors have not yet published results for this test and the Gabor, LBP and GMRF implementations are not rotation-invariant.

4. CONCLUSIONS

We have introduced a new texture feature based on the sieve which is an invertible transform. In a similar manner to a wavelet transform, the new method produces images that are differences between successive filterings of the input image. The sieve is known to be quick to compute [1] so the feature can be computed efficiently, certainly at video rate, and these experiments show that its performance is comparable to other methods. Since the method uses statistics from connected sets it is invariant to the rotation angle of the texture.

As the scale of the texture changes, the area of the connected sets change so do the locations of the peaks and troughs in the feature vector. Hence, if needed, the feature vector can be scaled to be invariant to zoom also. If rotational variance is required then we recommend the one-dimensional variant of the sieve which we are currently investigating.

5. REFERENCES

[1] J. Bangham, K. Moravec, R. Harvey, and M. Fisher. Scale-space trees and applications as filters for stereo vision and image retrieval. In *British Machine Vision Conference*, pages 113–123, 1999.

[2] J.A. Bangham, R. Harvey, P.D. Ling, and R.V. Aldridge. Morphological scale-space preserving transforms in many dimensions. *Journal of Electronic Imaging*, 5:283–299, 1996.

[3] P. Brodatz. *Textures : A Photographic Album*. New York : Dover, 1966.

[4] O.J. Cula and K.L. Dana. Compact representation of bidirectional texture functions. In *IEEE International Conference on Computer Vision and Pattern Recognition. (CVPR)*, 2001.

[5] K.J. Dana, S.K Van-Ginneken, and J.J. Koenderink. Reflectance and texture of real world surfaces. *ACM transactions on Graphics*, 43(1):1–34, 1999.

[6] B. Manjunath and W. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, 1996.

[7] T. Ojala, T. Maenpaa, M. Pietikainen, J. Viertola, J. Kyllonen, and S. Huovinen. Outex - new framework for empirical evaluation of texture analysis algorithms. In *Proc. 16th International Conference on Pattern Recognition, Quebec, Canada*, volume 1, pages 701–706, 2002.

[8] T. Ojala, M. Pietikanien, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29:51–59, 1996.

[9] M. Pietikainen, A. Rosenfeld, and L.S. Davis. Experiments with texture classification using averages of local pattern matches. *IEEE Trans. Systems, Man and Cybernetics*, 13(3):421–426, 1983.

[10] G. Smith and I. Burns. Measuring texture classification algorithms. *Pattern Recognition Letters*, 18:1495–1501, 1997.

[11] J. Strand and T. Taxt. Local frequency features for texture classification. *Pattern Recognition*, 27(10):1397–1406, 1999.

[12] M. Varma and A. Zisserman. Classifying images of materials: Achieving viewpoint and illumination independence. In *7th Eurp Conference on Computer Vision, Copenhagen, Denmark*, volume 3, pages 255–271, 2002.

[13] Vistex. http://www-white.media.mit.edu/vismod/imagery/visiontexture/vistex.html.