Texture Classification Using N-Tuple Pattern Recognition

L.Hepplewhite & T.J.Stonham
Department of Electronics and Electrical Engineering
Brunel University, Uxbridge, Middlesex, UB8 3PH, U.K.
Email: Lee.Hepplewhite@brunel.ac.uk

Abstract

This paper presents a novel approach to real-time texture classification, derived from the n-tuple method of Bledsoe & Browning, for use in industrial applications. In recent years, various approaches have been presented for the texture classification problem. However, few have the computational tractability needed in an automated environment. In this paper, methods for texture classification based on approximations to the nth order co-occurrence spectrum are discussed. Limitations of these methods are highlighted before a new method based around Marr's zero crossing sketch is presented. Preliminary results are presented comparing the new method and other n-tuple based schemes.

1. Introduction

Texture classification remains a fundamental task for image processing. A wealth of texture recognition methods are currently available [1][2], however few have the computational tractability needed in an automated environment. In this paper approaches based on n-tuple pattern recognition are discussed. N-tuple pattern recognition has achieved some success in real-time pattern recognition tasks [3] and was extended to texture recognition by Patel and Stonham [4] in the Binary Texture Co-occurrence Spectrum.

2. Binary Texture Co-occurrence Spectrum (BTCS)

In the BTCS textural information is represented at two distinct scales; that of micro-texture and that of macro-texture. Micro-texture information is extracted from an image in n pixel samples, termed n-tuples, using local neighbourhood operators (see fig. 1).

Since the BTCS is used to represent binary textures, each n-tuple represents a binary state in the range 0 to 2^n-1. Differences in the occurrence of these n-tuple states constitutes differences in texture at the micro-texture scale. However, differences in the relative occurrence of these states over a textural region can be used to describe texture at the macro-texture scale.

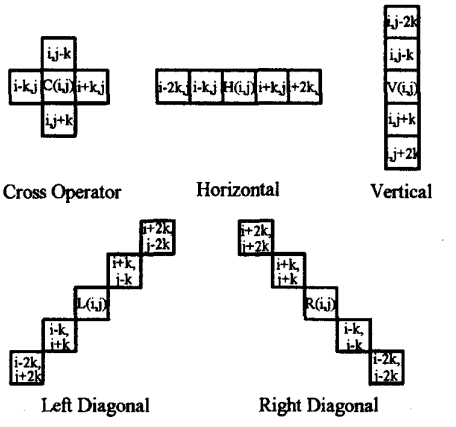

Fig.1 Local neighbourhood operators (4-tuple)

Hence the BTCS characterises a sample texture by recording the occurrence of these n-tuple states over the texture sample to form a 2^n dimensional state vector. These state vectors, termed the Binary Texture Co-occurrence Spectrum, can be readily sorted by conventional pattern recognition techniques as demonstrated in [4]. The simplicity and computational efficiency of this method make it an attractive solution to real-time texture classification. However real world textures are rarely binary in nature and the method's performance is restricted by the effectiveness of the thresholding techniques employed.

1015-4651/96 $5.00 © 1996 IEEE
Proceedings of ICPR '96

3. Grey Level Texture Co-occurrence Spectrum (GLTCS)

Unfortunately the BTCS does not scale up to the multivalued grey level case since a grey level image containing g intensity levels would produce a co-occurrence spectrum of dimensionality g^n . This necessitates some approximation of the n-tuple states to be made in order to reduce the dimensionality. Typically Rank coding is utilised [5][6][7] reducing the dimensionality to n! Again the method appears attractive to the industrial inspection environment as demonstrated by the authors in [8].

However, in reducing the dimensionality needed to represent the occurrence of the n-tuple states, rank coding is indiscriminant in the way it divides the pattern space; little or no attention is given to what textural features the rank codes are extracting. Since n-tuples are extracted by local neighbourhood operators, the grey levels changes within the n-tuples will occur gradually. This results in n-tuples extracting low order intensity profiles, typically 'edge' and 'bar' profiles. Unfortunately, as fig. 2 demonstrates, rank coding biases the coding towards higher order profiles which are less common in the data and more susceptible to noise.

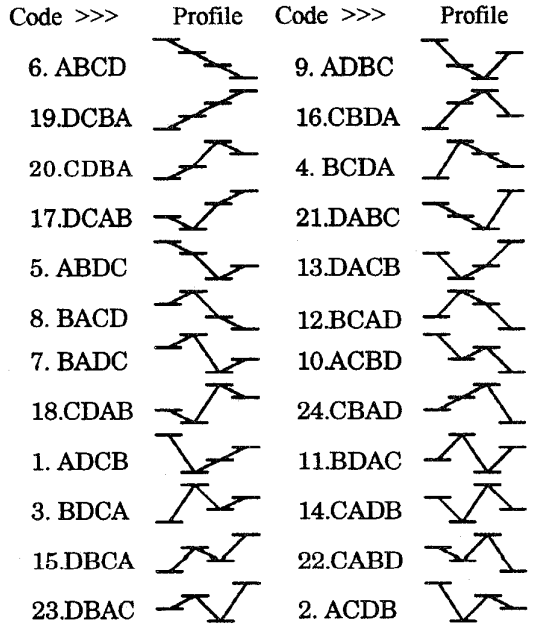

Fig.2 Rank coding of 4-tuples and the luminance profiles they represent

Thus low order profiles (RANK codes 6 & 19) dominate the co-occurrence spectrum resulting in all texture spectrums appearing similar for all textures (see Fig. 3). Wang and He[9] present a similar method, the Texture Unit and Texture Spectrum (TUTS), which attempts to code local 3x3 window profiles as texture units. Examination of the texture spectra produced by this method confirms that only a discrete set of the 6561 possible texture units actually occur in the textures. Again these correspond to 'edge' profiles at various orientations suggesting that in common with more elaborate texture description methods, texture discrimination is based on the spatial distribution of simple edge profiles at different spatial frequencies.

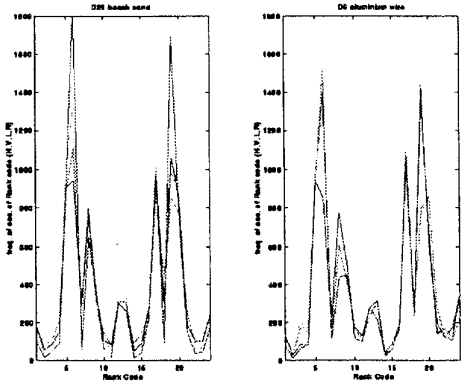

Fig. 3 GLTCS representation of Brodatz Textures appear similar

4. Zero Crossings Texture Co-occurrence Spectrum (ZCTCS)

There have been several methods for local frequency decomposition presented in the literature, i.e. Gabor, wavelet etc. Fortunately in an industrial application the environment is usually controlled and it is highly likely that only a small discrete set of scales will be important. By choosing the dominant scale of the textures under inspection, smoothing (low pass filtering) can be performed at that scale. Subsequently a gradient operator can be used to describe the edge features at that scale.

In this paper the filter chosen to perform these two tasks is the Laplacian of Gaussian (LoG) or Mexican Hat Filter (Eqn. 1) as used by Marr [10]. An important property of the filter is it's balance between positive and negative values. This means that the response to dark/light edges is exactly opposite to that of light/dark edges. The image resulting from convolution of this filter with a texture defines edges within the texture as zero crossings in the new image.

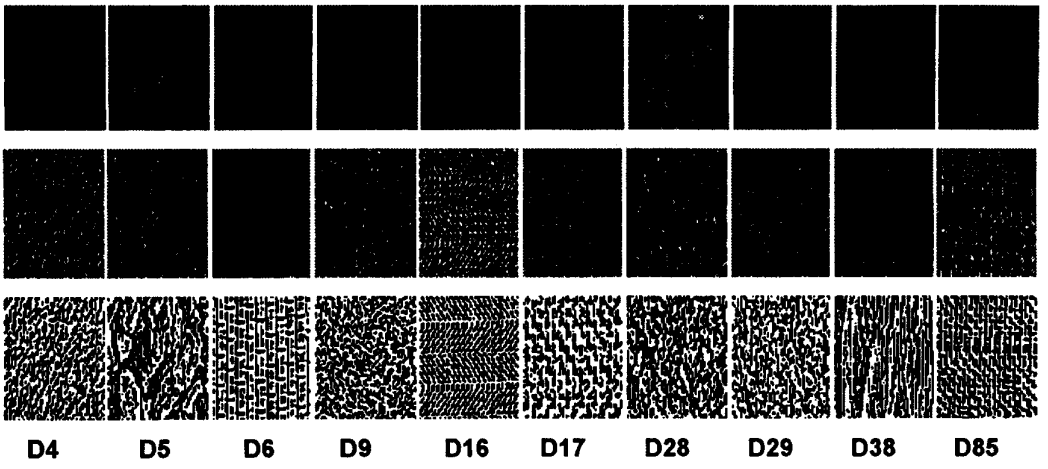

Fig.4 Brodatz textures , grey (top row), LoG filtered (middle), Binarized LoG (bottom)
Textures are (L to R) : pressed cork, expanded mica(x3), woven aluminium wire, grass, herringbone weave, herringbone weave(x4), beach sand(x4), beach sand, water, straw matting

$$\nabla^2 G(x, y) = \left(1 - \frac{x^2+y^2}{2\sigma^2}\right) e^{\frac{-(x^2+y^2)}{2\sigma^2}} \qquad (1)$$

In order to reduce the complexity of future processing the output of the filter is binarized. This binarization acts as a 'blob' detector whilst preserving the sign of the zero crossings present in the image. As can be seen in Fig.4 the grey level texture recognition problem has been reduced to a binary texture recognition problem which can be resolved by the BTCS. The resulting method retains the attractiveness of the BTCS whilst making it independent of threshold level at the small computational expense of a simple pre-processing operation.

5. Results

In order to demonstrate the effectiveness of the new method it's performance has been compared with the BTCS, GLTCS and the TUTS of Wang & He [9]. To evaluate each method's discriminating performance, classification has been performed on the set of ten brodatz textures shown in Fig.4 (top row). Ten sample windows were randomly chosen from each of the ten texture classes. Each of the samples was then transformed using the four methods under test. Following the transforms the resulting texture spectrums (state vectors) were classified using a 'leave-one-out' procedure and a simple Euclidean distance classifier[11].

Table 1 shows the classification results for various interpixel spacings. A window size of 64 pixels was chosen as it effectively extracts typical archetypes for the textures. The texture micrographs were 256 x 256 pixels x 256 grey level images taken from the brodatz texture album [12]. In the BTCS, GLTCS and the ZCTCS methods, four 4-tuple operators, horizontal, vertical, right and left diagonals were used to form 64, 96 and 64 dimensional texture spectrums respectively. To binarize the textures a global threshold level of 80 was used in the BTCS method. The TUTS method was implemented as in [9] using a 3x3 operator and a 6561 dimensional texture spectrum. In the ZCTCS the standard deviation of the filter, σ, was chosen to match the dominant scale of the majority of the texture set (σ = 1). This defines the width of the excitatory region of the mexican hat filter and thus defines the optimal interpixel spacing for the n-tuple operator.

	k=1	k=2	k=3	miss-classified textures
BCTS	79%	-	-	2,3,7,8,9
TUTS	97%	-	-	6,7,8
GLTCS	98%	97%	93%	7,8,9
ZCTCS	100%	100%	97%	2,7
ZCTCS (cross op)	95%	100%	92%	2,7

Table 1 Result of classification

Also shown in the results is the performance of the ZCTCS with a simple cross operator instead of the four orientational masks. As the results demonstrate the performance of the ZCTCS is comparable with others methods. However the important advantage of the method is in which samples are miss-classified. Any misclassifications made by the ZCTCS are exclusively associated with classes 2 and 7 (D5 and D28). This is a result of the scale of the LoG filter being chosen to discriminate the other, finer scale textures. In order to classify all textures, another filter would be needed to discriminate D5 and D28.

The improved performance of the ZCTCS over the GLTCS can be explained on inspection of the co-occurrence spectra associated with each method. The ZCTCS clearly transforms each texture to a distinct cluster in feature space, whereas the GLTCS clusters all textures in the same region of feature space as demonstrated in fig. 3. Using Fisher's linear discriminant criterion [11], the relative performance of the ZCTCS and GLTCS has been investigated.

A measure of the separation of two classes is the difference of their sample means. We define m_i as the sample mean for class i, given by:

$$m_i = \frac{1}{n_i} \sum_{x \in \chi_i} x \qquad (2)$$

where x is a feature vector representation of a texture

χ_i is set of all vectors, x, which belong to class i

n_i is the cardinality of χ_i

It follows that the difference between sample means of classes i and j, in Euclidean distance sense, is given by:

$$D(i,j) = |m_i - m_j| \qquad (3)$$

In order to obtain good separation between classes this difference must be maximised relative to some measure of the standard deviations for each class. In Fisher's linear discriminant, the within class scatter is defined:

$$s_i^2 = \sum_{x \in \chi_i} (x - m_i)^2 \qquad (4)$$

Combining the two measures, Fisher's criterion is defined as:

$$J(i,j) = \frac{|m_i - m_j|^2}{s_i^2 + s_j^2} \qquad (5)$$

The goal of each texture transform is to map each texture to a unique cluster in feature space. This is achieved by maximising the separation between classes whilst minimising the within class scatter. These criteria are represented by maximising the Fisher criterion derived above. The performance of the GLTCS and ZCTCS is demonstrated in tables 3 and 4 respectively.

As can be clearly seen in the tables, the ZCTCS improves on the GLTCS for the majority of textures used at the expense of reducing discrimination between textures two and seven (D5 & D28). As already explained this is a result of the mexican hat filter being tuned to the dominant high spatial frequencies in the other textures. Since a higher dimensionality of the feature space would result in greater scope for class representation, the result has increased significance as the ZCTCS has a lower dimensionality than that of the GLTCS.

class i									
2	0.59								
3	0.36	0.47							
4	0.15	0.6	0.42						
5	1.09	1.76	1.73	1.02					
6	0.49	0.17	0.38	0.72	2.24				
7	0.35	0.13	0.32	0.47	1.73	0.07			
8	0.14	0.34	0.27	0.25	1.66	0.25	0.16		
9	0.3	0.23	0.17	0.38	1.01	0.1	0.12	0.24	
10	0.27	0.57	0.43	0.15	0.93	0.67	0.53	0.25	0.48
class j	1	2	3	4	5	6	7	8	9

Table 2 Fisher's Criterion for the GLTCS
(class numbers correspond to Fig.4)

class i									
2	0.33								
3	3.28	1.44							
4	0.35	0.37	2.8						
5	2.73	1.39	7.56	1.87					
6	1.1	0.47	0.61	1.03	3.67				
7	0.39	0.07	2.95	0.38	2.56	0.91			
8	0.61	0.4	1.35	0.53	3.21	0.32	0.53		
9	0.69	0.28	1.49	0.96	2.95	0.61	0.41	0.44	
10	0.88	0.3	3.4	0.71	3.45	1.4	0.3	0.95	0.53
class j	1	2	3	4	5	6	7	8	9

Table 3 Fisher's Criterion for the ZCTCS
(class numbers correspond to Fig.4)

6. Conclusions

The new method has been shown to extend the usefulness of n-tuple pattern recognition methods for texture classification. Results presented demonstrate how the n-tuple methods can be 'tuned' to certain spatial frequencies in the texture. Due to the low dimensionality of the method's representation, particularly the cross operator, several spatial frequencies could be described at once without degrading performance significantly.

Current research on texture assumes some form of frequency decomposition of the input followed by a non-linear operator as the first stage of processing (e.g. LoG, Gabor etc.). The question now is how to represent the next level of processing. In this paper we present a simplistic, engineering solution which makes real-time texture recognition possible.

Acknowledgements

The authors acknowledge funding of this research by Xyratex / IBM Havant U.K They also acknowledge work carried out at Brunel by Alexander Geist as part of the Erasmus Exchange.

References

[1] L.Van Gool, P.Dewaele & A.Oosterlinck, Texture Analysis Anno 1983, CVGIP Vol.29 p.336-357

[2] M.Tuceryan & A.K.Jain, Texture Analysis, Handbook of Pattern Recognition and Computer Vision, chp. 2.1, p.235-276

[3] I.Aleksander & T.J.Stonham, Guide to pattern recognition using random-access memories, Computers and Digital Techniques, vol. 2, no. 1, p.29-40

[4] D.Patel & T.J.Stonham, A single layer neural network for texture discrimination, Proc. IEEE International Symposium on circuits and systems, Singapore, 1991, p. 2657

[5] J.Austin, Grey level n-tuple processing, Proc. 4th ICPR, p.28, 1988

[6] D.Patel & T.J.Stonham, Unsupervised / supervised texture segmentation and its applications to real world data, SPIE Visual Communications & Image Processing vol.1818, Boston 1992, p.1206

[7] L.Hepplewhite & T.J.Stonham, Surface Inspection using Texture Recognition, Proc. 12th ICPR, vol.A, p.589-591

[8] L.Hepplewhite & T.J.Stonham, Inspecting the quality of magnetic disks, Image Processing, Vol.7 No.1, p.24-26

[9] L.Wang & D.C.He , Texture Classification using the Texture Spectrum, Pattern Recognition, Vol.23, No.8, p.905, 1990

[10] D.Marr, Vision, W.H.Freeman & co. 1982, ISBN 0-7167-1567-8

[11] R.O.Duda & P.E.Hart, Pattern Classification and Scene Analysis, NY:Wiley,1973

[12] P.Brodatz, Texture: A photographic album for artists and designers, Dover, ISBN 0-486-21669-1