EVALUATION OF TEXTURE METHODS FOR IMAGE ANALYSIS

Mona Sharma
Markos Markou
Sameer Singh*

{m.sharma, m.markou, s.singh}@exeter.ac.uk

PANN Research
Department of Computer Science
University of Exeter
Exeter EX4 4PT
United Kingdom

*Correspondence: Dr. Sameer Singh, Department of Computer Science, University of Exeter, Exeter EX4 4PT, UK; tel: +44-1392-264053; fax: +44-1392-264067;

EVALUATION OF TEXTURE METHODS FOR IMAGE ANALYSIS

ABSTRACT

The evaluation of texture features is important for several image processing applications. Texture analysis forms the basis of object recognition and classification in several domains. There is a range of texture extraction methods and their performance evaluation is an important part of understanding the utility of feature extraction tools in image analysis. In this paper we evaluate five different feature extraction methods. These are auto-correlation, edge frequency, primitive-length, Law's method, and co-occurrence matrices. All these methods are used for texture analysis of Meastex database. This is a publicly available database and therefore a meaningful comparison between the various methods is useful to our understanding of texture algorithms. Our results show that the Law's method and co-ccurrence matrix method yield the best results. The overall best results are obtained when we use features from all five methods. Results are produced using leave-one-out method.

Keywords: Image analysis, texture, feature extraction, benchmark, Meastex, performance analysis

1. INTRODUCTION

The understanding of synthetic and natural image objects is an important part of image understanding studies. The interpretation of images is only possible if classifiers can effectively label previously unseen objects. The recognition ability of classifiers depends on the quality of feature used as well as the amount of training data available to them. Image features are mostly extracted on shape and texture of segmented objects. Karu et al. [5] define a methodology for automatically identifying textured regions within an image so that feature extraction algorithms are only used where texture can be quantified. The coarseness parameter for detecting texture can be adjusted to find appropriate texture order.

Texture analysis has been used in a range of studies for recognising synthetic and natural textures. However, the number of studies on performance evaluation of various methods remains small. Previous studies have either compared too few texture methods or on small number of samples for any meaningful conclusion (see [15] for a criticism of various studies on performance evaluation). Each study has used a different combination of texture methods. Texture methods used can be categorised as: statistical, geometrical, structural, model-based and signal processing features [17]. Van Gool et al. [18] and Reed and Buf [13] present a detailed survey of the various texture methods used in image analysis studies. Randen and Husoy [12] conclude that most studies deal with statistical, model-based and signal processing techniques. Weszka et al. [19] compared the Fourier spectrum, second order gray level statistics, co-occurrence statistics and gray level run length statistics and found the co-occurrence were the best. Similarly, Ohanian and Dubes [8] compare Markov Random Field parameters, multi-channel filtering features, fractal based features and co-occurrence matrices features, and the co-occurrence method performed the best. The same conclusion was also drawn by Conners and Harlow [2] when comparing run-length difference, gray level difference density and power spectrum. Buf et al. [1] however report that several texture features have roughly the same performance when evaluating co-occurrence features, fractal dimension, transform and filter bank features, number of gray level extrema per unit area and curvilinear integration features. Compared to filtering features [12], co-occurrence based features were found better as reported by Strand and Taxt [14], however, some other studies have supported exactly the reverse. Pichler et al. [10] compare wavelet transforms with adaptive Gabor filtering feature extraction and report superior results using Gabor technique. However, the computational requirements are much larger than needed for wavelet transform, and in certain applications accuracy may be compromised for a faster algorithm. Ojala et al. [9] compared a range of texture methods using nearest neighbour classifiers including gray level difference method, Law's measures, center-symmetric covariance measures and local binary patterns applying them to Brodatz images. The best performance was achieved for the gray level difference method. Law's measures are criticised for not being rotationally invariant, for which reason other methods performed better.

In this paper we analyse the performance of five popular texture methods on the publicly available Meastex database [7,15]. Our primary aim is to understand which of the five methods is most suited to the texture recognition problem and whether the combination of features from different methods provides any further advantage.

2. TEXTURE METHODS

In this paper we analyse Meastex images using five different texture extraction methods. These methods are described in brief below [11,16].

Autocorrelation based texture features

The textural character of an image depends on the spatial size of texture primitives. Large primitives give rise to coarse texture (e.g. rock surface) and small primitives give fine texture (e.g. silk surface). An autocorrelation function can be evaluated that measures this coarseness. This function evaluates the linear spatial relationships between primitives. If the primitives are large, the function decreases slowly with increasing distance whereas it decreases rapidly if texture consists of small primitives. However, if the primitives are periodic, then the autocorrelation increases and decreases periodically with distance. The set of autocorrelation coefficients shown below are used as texture features:

$$C_{ff}(p,q) = \frac{MN}{(M-p)(N-q)} \frac{\sum_{i=1}^{M-p} \sum_{j=1}^{N-q} f(i,j) f(i+p, j+q)}{\sum_{i=1}^{M} \sum_{j=1}^{N} f^2(i,j)}$$

, where p, q is the positional difference in the i, j direction, and M, N are image dimensions. In this study we vary (p,q) from (0,0) to (9,9) giving us a total of 100 features.

Co-ccurrence matrices texture features

Statistical methods use second order statistics to model the relationships between pixels within the region by constructing Spatial Gray Level Dependency (SGLD) matrices [3]. A SGLD matrix is the joint probability occurrence of gray levels i and j for two pixels with a defined spatial relationship in an image. The spatial relationship is defined in terms of distance d and angle θ. If the texture is coarse and distance d is small compared to the size of the texture elements, the pairs of points at distance d should have similar gray levels. Conversely, for a fine texture, if distance d is comparable to the texture size, then the gray levels of points separated by distance d should often be quite different, so that the values in the SGLD matrix should be spread out relatively uniformly. Hence, a good way to analyse texture coarseness would be, for various values of distance d, some measure of scatter of the SGLD matrix around the main diagonal. Similarly, if the texture has some direction, i.e. is coarser in one direction than another, then the degree of spread of the values about the main diagonal in the SGLD matrix should vary with the direction d. Thus texture directionality can be analysed by comparing spread measures of SGLD matrices constructed at various distances d. From SGLD matrices, a variety of features may be extracted. The original investigation into SGLD features was pioneered by Haralick et al. [4]. From each matrix, 14 statistical measures are extracted including: angular second moment, contrast, correlation, variance, inverse different moment, sum average, sum variance, sum entropy, difference variance, difference entropy, information measure of correlation I, information measure of correlation II, and maximal correlation coefficient. The measurements average the feature values in all four directions.

Edge-frequency based texture features

A number of edge detectors can be used to yield an edge image from an original image. We can compute an edge dependent texture description function E as follows:

$$E = |f(i,j) - f(i+d,j)| + |f(i,j) - f(i-d,j)| + |f(i,j) - f(i,j+d)| + f(i,j) - f(i,j-d)|$$

This function is inversely related to the autocorrelation function. Texture features can be evaluated by choosing specified distances d. We vary the distance d parameter from 1 to 70 giving us a total of 70 features.

Primitive length texture features

Coarse textures are represented by a large number of neighbouring pixels with the same gray level, whereas a small number represents fine texture. A primitive is a continuous set of maximum number of pixels in the same direction that have the same gray level. Each primitive is defined by its gray level, length and direction. If we were to represent $B(a,r)$ as the number of primitives of all directions having length r and gray level a, let M, N be image dimensions, L the number of gray levels, N_r the maximum primitive length in the images, and K the total number of runs given by $\sum_{a=1}^{L} \sum_{r=1}^{N_r} B(a,r)$, then we can define the following 5 features defining image texture:

Short primitive emphasis: $\frac{1}{K} \sum_{a=1}^{L} \sum_{r=1}^{N_r} \frac{B(a,r)}{r^2}$; Long primitive emphasis: $\frac{1}{K} \sum_{a=1}^{L} \sum_{r=1}^{N_r} B(a,r) r^2$

Gray level uniformity: $\frac{1}{K}\sum_{a=1}^{L}[\sum_{r=1}^{N_r}B(a,r)r^2]^2$; Primitive length uniformity: $\frac{1}{K}\sum_{a=1}^{L}[\sum_{r=1}^{N_r}B(a,r)]^2$

Primitive percentage: $\frac{K}{\sum_{a=1}^{L}\sum_{r=1}^{N_r}rB(a,r)}=\frac{K}{MN}$

Law's texture features

Laws [6] observed that certain gradient operators such as Laplacian and Sobel operators accentuated the underlying microstructure of texture within an image. This was the basis for a feature extraction scheme based a series of pixel impulse response arrays obtained from combinations of 1-D vectors shown in Figure 1. Each 1-D array is associated with an underlying microstructure and labelled using an acronym accordingly. The arrays are convolved with other arrays in a combinatorial manner to generate a total of 25 masks, typically labelled as *L5L5* for the mask resulting from the convolution of the two L5 arrays.

Figure 1. Five 1-D arrays identified by Laws [6].

Level	L5 =	[	1	4	6	4	1	]
Edge	E5 =	[	-1	-2	0	2	1	]
Spot	S5 =	[	-1	0	2	0	-1	]
Wave	W5 =	[	-1	2	0	-2	1	]
Ripple	R5 =	[	1	-4	6	-4	1	]

These masks are subsequently convolved with a texture field to accentuate its microstructure giving an image from which the energy of the microstructure arrays is measured together with other statistics. We compute 5 amplitude features mean, standard deviation, skewness, kurtosis, and energy measurements. Since there are 25 different convolutions, altogether we obtain a total of 125 features.

For all feature extraction methods, we select the most appropriate features from classification using a linear stepwise discriminant analysis. This technique eliminates any features that are weak in their ability to differentiate between different classes. The final set of features are used for all further analysis.

3. IMAGE DATA

In this study we use the publicly available database Meastex. These sets contain examples of artificial and natural textures. The natural images have been chosen for their homogeneity of texture. Each image has a size of 512x512 pixels and is distributed in raw PGM format. Each image is split into 16 sub-images to increase the number of samples available for each class. The textures are available for classes asphalt, concrete, grass and rock. Finally we get a total of 944 images from which texture features are extracted. Details of the images available for each class are shown in Table 1.

Table 1. Samples for different image categories in Meastex database

Label	Class	Images
1	Asphalt	64
2	Concrete	192
3	Grass	288
4	Rock	400

The data for these classes is overlapping no matter what feature extraction method is employed. Therefore, their classification is not a trivial task.

4. PERFORMANCE EVALUATION

The performance evaluation of the five texture methods is based on the ability of a classifier to recognise unseen samples of the four classes on the basis of training data. The texture method that generates the best classification performance is ranked the best. In our analysis, we use a linear method of classification (Linear Discriminant Analysis), and two modified k-nearest neighbour methods for analysis. The algorithms of the nearest neighbour method are given in the appendix. In the nearest neighbour approach, we can set the number of nearest neighbours, parameter k, equal to an odd number. We use k=1, 3, 5, and 7. Our analysis is based on the leave one out method. Using this, for N samples altogether, we take out one sample at a given time for testing, and use the rest as training data. The process iterates a total of N times, each time with a different test sample and training set, and the recognition rate is averaged over N trials.

We first show the results of k-nearest neighbour (model 1) on our data set. Table 2 shows that the best results are obtained for the co-occurrence features, followed by the auto-correlation method, Law's method, Edge frequency method, and finally primitive length method. In the same table for the second nearest neighbour model, we get similar results except for the fact that the edge frequency method outperforms the Law's method. In Table 2, the recognition rates for the co-occurrence matrices are very high. The variability in the performance of the texture methods is remarkable and noteworthy. In Table 3, we show the results of the linear classification. The linear method gives excellent recognition rate showing that data is mostly linearly separable for better quality texture methods such as the co-occurrence approach and Law's method. We next combine features from all four methods as previous studies have indicated better performances when combining features from more than one approach [9]. A stepwise linear analysis is used to eliminate any feature that does not contribute to the discriminatory process. The remaining features are used for the classification purposes. The aim of this experiment is to see whether combining texture features from different methods is any superior to the best performing single texture method. The results are shown in Table 4. In Figure 2, we plot the best performance for each texture feature extraction method on Meastex data independent of the model used for nearest neighbour method. Figure 3 shows a similar plot for the Discriminant analysis.

Table 2. Classification accuracy of k-nearest neighbour method (models 1 and 2), and Discriminant Analysis on Meastex. The best results for each texture method and overall best has been highlighted.

Classifier	Texture Method	*k*=1	*k*=3	*k*=5	*k*=7
k-NN (model 1)	*Autocorrelation*	77.7%	79.3%	**79.4%**	79.3%
k-NN (model 2)	*Autocorrelation*	**77.7%**	75.9%	70.8%	68.9%
k-NN (model 1)	*Co-occurrence*	86.6%	86.0%	**86.8%**	86.5%
k-NN (model 2)	*Co-occurrence*	**86.6%**	84.8%	80.1%	77.2%
k-NN (model 1)	*Edge Frequency*	69.8%	**70.7%**	69.1%	70.1%
k-NN (model 2)	*Edge Frequency*	**69.8%**	64.7%	62.0%	59.4%
k-NN (model 1)	*Primitive Length*	45.2%	49.7%	52.7%	**54.1%**
k-NN (model 2)	*Primitive Length*	**45.2%**	42.5%	40.7%	40.2%
k-NN (model 1)	*Law's*	67.8%	72.2%	73.3%	**75.1%**
k-NN (model 2)	*Law's*	**67.8%**	64.7%	60.5%	56.6%

Table 3. Classification accuracy of Linear Discriminant Analysis on Meastex. The best results for each texture method and overall best has been highlighted.

Texture Method	Recognition rate
Autocorrelation	76.1%
Co-occurrence	79.2%
Edge Frequency	63.4%
Primitive Length	43.1%
Law's	82.8%

Figure 2. Plot of the recognition rate for each texture method for the best nearest neighbour classifier using leave one out validation procedure.

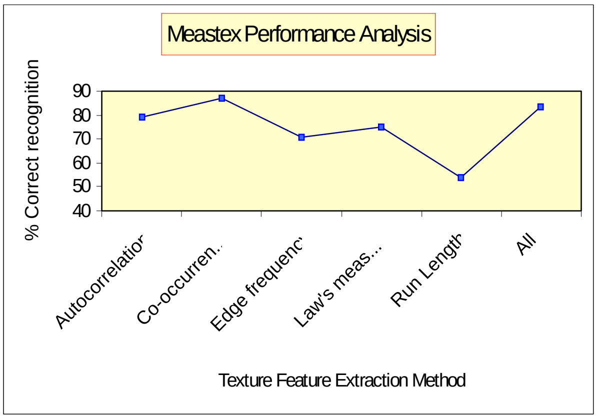

Figure 3. Plot of the recognition rate for each texture method for the Linear Discriminant Analysis using leave one out validation procedure.

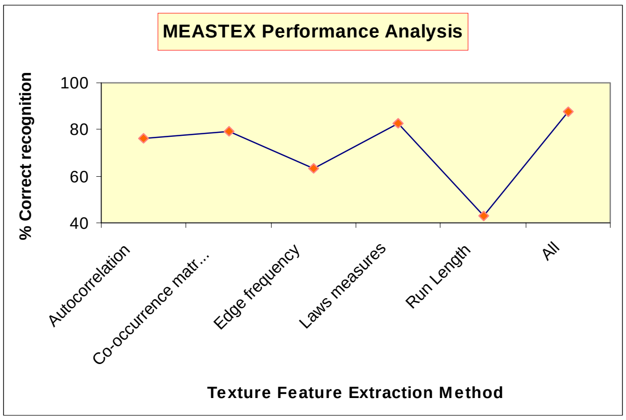

Table 4. Classification accuracy of k-nearest neighbour method (models 1 and 2) on Meastex data using combined features. The best performance is highlighted.

Classifier	***k*=1**	***k*=3**	***k*=5**	***k*=7**
Model 1	80.9%	82.6%	**83.3%**	80.5%
Model 2	80.9%	78.0%	75.5%	72.9%

The confusion matrices for the combined set of features is shown in Table 5. The confusion matrix shown here is similar in content to those obtained using individual feature extraction methods. As a general trend, asphalt and concrete samples are confused as rock samples, and vice-versa. There is also some overlap between rock and grass samples.

Table 5. Confusion Matrices based on nearest neighbour and discriminant analysis of on MEASTEX data using Leave-one-out validation method.

Model 1 - *k*=5

class	**1**	**2**	**3**	**4**
1	**37**	2	0	25
2	5	**159**	0	28
3	0	0	**270**	18
4	14	36	29	**321**

83.3% classification

Model 2 - *k*=1

class	**1**	**2**	**3**	**4**
1	**33**	3	0	28
2	5	**159**	0	28
3	0	0	**266**	22
4	26	30	37	**307**

80.9% classification

Linear discriminant analysis

class	**1**	**2**	**3**	**4**
1	**64**	0	0	0
2	6	**180**	0	6
3	0	0	**278**	10
4	40	42	12	**290**

87.5% classification

In the case of nearest neighbour models, the combined set of features do not improve the result. Co-occurrence matrices on their own have the best performance of 86.8% correct recognition. On the other hand when we use Linear Discriminant Analysis on combined features, we obtain the result of 87.5% correct classification that is better than any other result so far. Thus, there is a convincing argument that combination of features significantly increases the ability of the classifier to recognise texture in images.

5. CONCLUSION

In this paper we have used five different texture feature extraction methods that are most popularly used in image understanding studies. One of the features of this study is the use of a publicly available benchmark that further studies can use. Our results show that there is considerable performance variability between the various texture methods. Our finding, that co-occurrence matrices and Law's method perform better than other techniques, is supported by previous comparative studies in this area. It is however difficult to generalise this for all cases. The difference in results between the linear analysis and nearest neighbour method is also noteworthy. The best overall result using nearest neighbour methods is obtained with co-occurrence matrices, whereas using linear analysis the best result is obtained using combined set of features. It appears that since different texture methods capture different aspects of the image texture, and combining features from them has certainly much utility.

REFERENCES

[1] J.M.H. Buf, M. Kardan and M. Spann, "Texture feature performance for image segmentation", Pattern Recognition, vol. 23, no. 3/4, pp. 291-309, 1990.
[2] R.W. Conners and C.A. Harlow, "A theoretical comparison of texture algorithms", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 2, no. 3, pp. 204-222, 1980.
[3] J.F. Haddon, J.F. Boyce, Co-occurrence matrices for image analysis, IEE Electronics and Communications Engineering Journal, vol. 5, No 2, pp. 71-83, 1993.
[4] R. M. Haralick, K. Shanmugam and I. Dinstein, Textural features for image classification, IEEE Transactions on System, Man, Cybernetics, vol. SMC-3, pp. 610-621, 1973.
[5] K. Karu, A.K. Jain and R.M. Bolle, "Is there any texture in the image?", Pattern Recognition, vol. 29, no. 9, pp. 1437-1446, 1996.
[6] K.I. Laws, *Textured image segmentation*, PhD Thesis, University of Southern California, Electrical Engineering, January 1980.
[7] Meastex database: http://www.cssip.elec.uq.edu.au/~guy/meastex/meastex.html
[8] P.P. Ohanian and R.C. Dubes, "Performance evaluation for four class of texture features", Pattern Recognition, vol. 25, no. 8, pp. 819-833, 1992.
[9] T. Ojala, M. Pietikainen, "A comparative study of texture measures with classification based on feature distributions, Pattern Recognition, vol. 29, no. 1, pp. 51-59, 1996.
[10] O. Pichler, A. Teuner and B.J. Hosticka, "A comparison of texture feature extraction using adaptive Gabor filter, pyramidal and tree structured wavelet transforms", Pattern Recognition, vol. 29, no. 5, pp. 733-742, 1996.
[11] W.K. Pratt, *Digital image processing*, John Wiley, New York, 1991.
[12] T. Randen and J.H. HusØy, "Filtering for texture classification: A comparative study", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 4, pp. 291-310, 1999.
[13] T. R. Reed and J.M.H. Buf, "A review of recent texture segmentation and feature extraction techniques", Computer Vision, Image Processing and Graphics, vol. 57, no. 3, pp. 359-372, 1993.
[14] J. Strand and T. Taxt, "Local frequency features for texture classification", Pattern Recognition, vol. 27, no. 10, pp. 1397-1406, 1994.
[15] G. Smith and I. Burns, "Measuring texture classification algorithms", Pattern Recognition Letters, vol. 18, pp. 1495-1501, 1997.
[16] M. Sonka, V. Hlavac and R. Boyle, *Image processing, analysis and machine vision*, PWS publishing, San Francisco, 1999.
[17] M. Tuceyran and A.K. Jain, "Texture analysis", Handbook of Pattern Recognition and Computer Vision, C.H. Chen, L.F. Pau and P.S.P. Wang (Eds.), chapter 2, pp. 235-276, World Scientific, Singapore, 1993.
[18] L. vanGool, P. Dewaele and A. Oosterlinck, "Texture analysis", Computer Vision, Graphics and Image Processing, vol. 29, pp. 336-357, 1985.
[19] J.S. Weszka, C. R. Dyer and A. Rosenfeld, "A comparative study of texture measures for terrain classification", IEEE Transactions on Systems, man and Cybernetics, vol. 6, pp. 269-285, 1976.

Appendix

Nearest Neighbour Classifier Algorithms

Model-1 NN rule

- Out of n training vectors, identify the k nearest neighbours, *irrespective* of class label. k is chosen to be odd.
- Out of these k samples, identify the number of vectors, k_i, that belong to class ω_i, i=1, 2, …M. Obviously $\sum_i k_i = k$.
- Assign x to the class ω_i with the maximum number k_i of samples.
- If two or more classes ω_i, $i \in [1 \ldots M]$, have an equal number E of maximum nearest neighbours, then we have a tie (*conflict*). Use conflict resolution strategy.
- For each class involved in the conflict, determine the distance d_i between test pattern $x = \{x_1, \ldots x_N\}$ and class ω_i based on the E nearest neighbours found for class ω_i. If the m^{th} training pattern of class ω_i involved in the conflict is represented as $y^{im} = \{y_1^{im}, \ldots y_N^{im}\}$ then the distance between test pattern x and class ω_i is: $d_i = \frac{1}{E} \sum_{j=1}^{N} |(x_j - y_j^{im})|$
- Assign x to class C if its d_i is the smallest, i.e. $x \in \omega_C$, if $d_C < d_i$ for $\forall i$, such that $C \in [1 \ldots M]$ and $i \neq C$.

Model-2 NN rule

- Out of n training vectors, identify the k nearest neighbours, *irrespective* of class label. k is chosen to be odd.
- Out of these k samples, identify the number of vectors, k_i, that belong to class ω_i, i=1, 2, …M. Obviously $\sum_i k_i = k$.
- Find the average distance d_i that represents the distance between test pattern $x = \{x_1, \ldots x_N\}$ and E_i nearest neighbours found for class ω_i, $i = 1 \ldots M$. Only include classes for which samples were detected in the first step. If the m^{th} training pattern of class ω_i found within the hypershere is represented as $y^{im} = \{y_1^{im}, \ldots y_N^{im}\}$, then the distance between test pattern x and class ω_i is: $d_i = \frac{1}{E_i} \sum_{j=1}^{N} |(x_j - y_j^{im})|$
- Assign x to class C if its d_i is the smallest, i.e. $x \in \omega_C$, if $d_C < d_i$ for $\forall i$, such that $C \in [1 \ldots M]$ and $i \neq C$. The decision in this model does not depend on the number of nearest neighbours found but solely on the average distance between the test pattern and samples of each class found.