Spatial Texture Analysis: A comparative Study

Maneesha Singh and Sameer Singh
PANN Research, Department of Computer Science,University of Exeter, Exeter, UK

Abstract

In this paper we compare some of the traditional, and some fairly new techniques of texture analysis on the MeasTex and VisTex benchmarks to illustrate their relative abilities. The methods considered include autocorrelation (ACF), co-occurrence matrices (CM), edge frequency (EF), Law's masks (LM), run length (RL), binary stack method (BSM), texture operators (TO), and texture spectrum (TS). In addition, we illustrate the advantage of using feature selection on a combined set that improves the overall recognition performance.

Keywords: texture, recognition rate, comparison, benchmark

1. Image Texture

The analysis of texture in images provides an important cue to the recognition of objects. In this paper, we have two objectives. First, to compare a range of traditional and fairly new methods of texture analysis on two popular benchmarks. Second, to investigate the advantage of selecting the best features from each technique to generate a feature set that gives better texture recognition performance.

Texture benchmark evaluation is not a new area of work, however previous work has either compared too few algorithms or used very small number of benchmark images that makes it difficult to generalise results (see [19] for a criticism of various studies on performance evaluation). Texture methods used can be categorised as: statistical, geometrical, structural, model-based and signal processing features [21]. Van Gool et al. [22] and Reed and Buf [16] present a detailed survey of the various texture methods used in image analysis studies. Randen and HusØy [15] conclude that most studies deal with statistical, model-based and signal processing techniques. Weszka et al. [24] compared the Fourier spectrum, second order grey level statistics, co-occurrence statistics and grey level run length statistics and found the co-occurrence were the best. Similarly, Ohanian and Dubes [11] 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 [3] when comparing run-length difference, grey 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 grey level extrema per unit area and curvilinear integration features. Compared to filtering features [15], co-occurrence based features were found better as reported by Strand and Taxt [18], however, some other studies have supported exactly the reverse. Pichler et al. [13] 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. [12] compared a range of texture methods using nearest neighbour classifiers. The best performance was achieved for the grey level difference method.

This work analyses the performance of eight popular texture methods on the publicly available Meastex database [10,19] and Vistex database [23]. The performance of the linear and kNN classifiers used is evaluated using leave-one-out cross-validated method. The paper is organised as follows. We first present details of the texture measures for data analysis. Meastex and Vistex databases are discussed in brief next followed by the experimental details. The results are finally discussed for the linear and nearest neighboour classifiers.

2. Spatial Texture Analysis

The texture extraction algorithms analyse the spatial distribution of pixels in grey scale images. The different methods capture how coarse or fine a texture is. The textural character of an image depends on the spatial size of texture primitives [7]. Large primitives give rise to coarse texture (e.g. rock surface) and small primitives give fine texture (e.g. silk surface). The eight feature extraction methods used here are based on this spatial element rather than analysing the frequency domain information of the given images. Their brief description appears below.

The *autocorrelation* method is based on finding 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 are computed by estimating the relationship between all pixel pairs f(x,y) and f(x+p, y+q), where the upper limit to the values of p and q is set by the user. The *co-occurrence* approach is based on the joint probability distribution of pixels in an image [4]. A co-occurrence matrix is the joint probability occurrence of grey 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 Thus texture directionality can be analysed by comparing spread measures of co-occurrence matrices constructed at various distances d. From co-occurrence matrices, a variety of features may be extracted. The original investigation into co-occurrence features was pioneered by Haralick et al. [5]. From each matrix, 20 statistical measures are extracted. For *edge frequency* method, we can compute the gradient difference between a pixel f(x,y) and its neighbours at a distance d. For a given value of distance, the gradient differences can be summed up over the whole image. For different values of d (in our case $1 \leq d \leq 50$), we obtain different feature measurements for the same image. For *Law's* method, a total of 25 masks are convolved with the image to detect different features such as linear elements, ripples, etc. These masks have been proposed by Law's [8]. We compute five amplitude features for each convolution, namely mean, standard deviation, skewness, kurtosis, and energy measurement. Finally, for *primitive length* features, we evaluate the number of strings of pixels that have the same grey level. Coarse textures are represented by a large number of neighbouring pixels with the same grey 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 grey level. Each primitive is defined by its grey level, length and direction. Five statistical features defining the characteristics of these primitives are used as our features. The detailed algorithms for these methods are presented by Sonka et al. [20] and Pratt[14].

In addition to the above well-known approaches to texture classification, we consider three new approaches including binary stack method, texture operators and texture spectrum. Chen et al. [2] introduced the use of binary stacks for texture analysis. For a total of L grey levels, L binary images are generated by thresholding the original image at each grey-level. The resulting stack of binary images is analysed by grouping all 1-and 0-valued pixels into connected regions. For each connected region, measures of irregularity or circularity are computed and weighted based on the total size of connected components. The characteristics computed include the number of 1-connected regions, number of 0-connected regions and two weighted irregularity measures. For these four characteristics, four statistical measures of range and spread are calculated as texture features. Manian et al. [9] presented a new algorithm for texture classification based on logical operators. These operators are based on order-2 elementary matrices whose building blocks are numbers 0, 1, and −1 and matrices of order 1x1. These matrices are operated on by operators such as row-wise join, column-wise join, etc. A total of six best operators are used and convolved with images to get texture features. Features are computed using zonal-filtering using zonal masks that are applied to the standard deviation matrix. Features obtained include horizontal and vertical slit features, ring feature, circular feature and sector feature. He and Wang[6] proposed the use of texture spectrum for extracting texture features. If an image can be considered to comprise of small texture units, then the frequency distribution of these texture units is a texture spectrum. The features extracted include black-white symmetry, geometric symmetry, degree of direction, orientation features and central symmetry.

3. Texture Benchmarks

MeasTex is a publicly available texture benchmark. Each image has a size of 512x512 pixels and is distributed in raw PGM format. We split each image into 16 sub-images to increase the number of samples available for each class. The textures are available for classes asphalt (64 samples), concrete (192 samples), grass (288 samples) and rock (400 samples). Images of type 'miscellaneous' have been excluded from this study. Finally we get a total of 944 images from which texture features are extracted.

All images in the Vision Texture (VisTex) database are stored as raw ppm (P6) files with a resolution of 512x512 pixels. The original VisTex database consists of 19 classes. Some of these classes have less than 5 sample images that have been removed from our analysis. Each original image was divided into 4 images to increase the number of available samples. We are finally left with 7 classes that are: bark (36 samples), fabric (80 samples), food (48 samples), metal (24 samples), sand (28 samples), tile (32 samples), and water (32 samples). Examples images of the MeasTex and VisTex benchmarks are shown in Figure 1.

The number of features extracted from each method are as follows: autocorrelation (99), co-occurrence matrices (20), edge frequency (50), Law's masks (125), run length (5), binary stack method (17), texture operators (30), and texture spectrum (7). For both MeasTex and VisTex, the pricipal component plots show strong overlaps across different classes. For each set of features, the principal components with eigen-values greater than one are used for further analysis. This allows us a more compact representation of data where features which capture the maximum variability of the overall set are used.

4. Experiments and Results

In this section we present the experimental details of MeaTex and VisTex data analysis. There are total of 944 samples for MeasTex data and 280 samples for VisTex data. We use leave-one-out method of cross-validation for exhaustively testing the data. In this method, for N samples, a total of N trials are conducted. In each trial a sample is taken out from the data set and kept for testing and the others are used for training. In each trial, therefore, we have a different set of training data and a different test data. The recognition performance is averaged across all trials. This methodology is superior to random partitioning of data to generate training and test set as the resultant performance of the system may not reflect its true ability for texture recognition.

MeasTex and VisTex Results

The results of MeasTex analysis are shown in Table 1. Clearly, the *k*NN classifier is far better as classification compared to the linear classifier. Although the binary stack method is superior using the linear classifier with respect to its nearest competing method, texture operators, we find that using the nearest neighbour classifier, the texture operator method gives the best performance. Both of these methods are better than others considered here. Except for co-occurrence matrices result, there is a wide margin in performance between these leading and other methods.

Texture Method	LDA	*k*NN k=1	*k*NN k=3	*k*NN k=5	*k*NN k=7	*k*NN k=9
ACF	76.1	79.3	78.2	77.4	77.5	78.8
CM	79.2	83.5	84.1	83.8	82.9	81.3
EF	63.5	69.0	69.0	69.3	69.7	71.3
LM	82.8	63.3	67.8	69.9	70.9	69.8
RL	43.1	45.3	46.1	46.5	51.1	51.9
BSM	82.8	92.9	93.1	93.0	91.9	91.2
TO	62.7	**94.6**	93.6	94.1	93.6	94.0
TS	61.0	68.3	67.3	67.9	68.5	68.1

Table 1. MeasTex Data Analysis: Average Leave-one-Out Recognition Rates

Texture Method	LDA	*k*NN k=1	*k*NN k=1	*k*NN k=1	*k*NN k=1	*k*NN k=1
ACF	72.1	87.1	78.6	76.4	66.4	65.0
CM	73.9	75.7	73.9	67.9	67.1	62.5
EF	53.2	69.3	70.4	67.1	65.7	62.9
LM	68.8	50.7	55.0	53.2	56.1	57.8
RL	34.8	36.8	36.1	36.1	38.6	40.0
BSM	77.9	78.2	73.6	68.6	66.1	63.6
TO	44.3	**93.2**	89.6	89.6	90.0	90.0
TS	40.0	37.9	37.2	35.7	33.5	33.6

Table 2. VisTex Data Analysis: Average Leave-one-Out Recognition Rates

On VisTex analysis shown in Table 2, as before the binary stack method is better than the texture operator method out of the two leading methods but the reverse becomes true on the use of nearest neighbour classifier. In this experiment, we get three clusters of performance. The leading method (texture operator), mediocre performances (binary stack method, autocorrelation, co-occurrence matrices, edge frequency, Law's masks), and poor performances (run length, texture spectrum). For the texture operator method of Manian et al.[9], the best performances of 94.6% and 93.2% correct recognition are very impressive compared to previously reported results on these benchmarks including our own reports [17].

Combined-data and Feature Selection Approach

The above comparative study helps us understand the relative strengths of eight texture analysis methods on two publicly available benchmarks. We next pool together the data from all texture feature sets and do feature selection based on those features that maximise the Mahalanobis distance. The sequential forward selection approach is followed.

Figure 2 shows the performance of the MeasTex and VisTex feature selection. A total of 48 features are pooled together based on the selected principal components of each method. On the combined set without feature selection, we get an overall recognition rate of 84.3% for MeasTex and 83.9% for VisTex. On MeasTex database, Sequential Forward Selection aimed at maximising the Mahalanobis metric does not improve the performance against the best single method (we get an overall best result of 94.2% correct by using 10 selected features from the pooled features, whereas in table 1 we found texture operator method to yield the best recognition rate of 94.6% which is slightly higher). It is interesting to note which 10 features from the pooled feature set were considered the best. These includes features from ACF(2), CM(3), EF(1), BSM(2) and TO(2). Next we consider feature selection for VisTex benchmark. We get a best recognition rate of 97.2% with 15 selected features which improves the previous best of 93.2% correct recognition using texture operators alone. It is interesting once more to note the composition of the selected 15 features. The features from the following algorithms are selected: ACF(2), CM(3), LM(2), EF(2), BSM(4), TO(2). It is interesting to note in both cases, run length features or texture spectrum features were note considered important. Also, the contribution of texture operator features in the VisTex best 15 features is not significant and a range of methods have balanced contribution.

5. Conclusion

We find that for both Meastex and Vistex data excellent results are obtained with the binary stack method and the texture operator method. The other feature extraction methods co-occurrence matrices, autocorrelation, Law's masks and edge frequency give similar but slightly inferior results. The run-length and texture spectrum performances are considerably poor. The performance of the linear classifier is fairly good but it improves considerably when using the nearest neighbour classifier. Also, we find that feature selection on pooled data gives the overall best performance.

References

[1] J.M.H. Buf, M. Kardan and M. Spann, Texture feature performance for image segmentation, *Pattern Recognition*, 23(3/4):291-309, 1990.

[2] Y.Q. Chen, M.S. Nixon and D.W. Thomas, Statistical geometrical features for texture classification, *Pattern Recognition*, vol. 28, no. 4, pp. 537-552, 1995.

[3] R.W. Conners and C.A. Harlow, A theoretical comparison of texture algorithms, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(3):204-222, 1980.

[4] J.F. Haddon, J.F. Boyce, Co-occurrence matrices for image analysis, *IEE Electronics and Communications Engineering Journal*, 5(2):71-83, 1993.

[5] R. M. Haralick, K. Shanmugam and I. Dinstein, Textural features for image classification, *IEEE Transactions on System, Man, Cybernetics*, 3:610-621, 1973.

[6] D.C. He and L. Wang, Texture features based on texture spectrum, *Pattern Recognition*, vol. 25, no. 3, pp. 391-399, 1991.

[7] K. Karu, A.K. Jain and R.M. Bolle, Is there any texture in the image? *Pattern Recognition*, 29(9):1437-1446, 1996.

[8] K.I. Laws, *Textured image segmentation*, PhD Thesis, University of Southern California, Electrical Engineering, January 1980.

[9]V. Manian, R. Vasquez and P. Katiyar, Texture classification using logical operators, *IEEE Transactions on Image Analysis*, vol. 9, no. 10, pp. 1693-1703, 2000.

[10] Meastex database:
http://www.cssip.elec.uq.edu.au/~guy/meastex/meastex.html

[11] P.P. Ohanian and R.C. Dubes, Performance evaluation for four class of texture features, *Pattern Recognition*, 25(8):819-833, 1992.

[12] T. Ojala, M. Pietikainen, A comparative study of texture measures with classification based on feature distributions, *Pattern Recognition*, 29(1):51-59, 1996.

[13] 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*, 29(5): 733-742, 1996.

[14] W.K. Pratt, *Digital image processing*, John Wiley, New York, 1991.

[15] T. Randen and J.H. HusØy, Filtering for texture classification: A comparative study, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):291-310, 1999.

[16] T. R. Reed and J.M.H. Buf, A review of recent texture segmentation and feature extraction techniques, *Computer Vision, Image Processing and Graphics*, 57(3):359-372, 1993.

[17] S. Singh and M. Sharma, Texture Analysis Experiments with Meastex and Vistex Benchmarks, Proc. International Conference on Advances in Pattern Recognition, Lecture Notes in Computer Science no. 2013, S. Singh, N. Murshed and W. Kropatsch (Eds.), Springer, pp. 417-424, Rio (11-14 March, 2001).

[18] J. Strand and T. Taxt, Local frequency features for texture classification, *Pattern Recognition*, 27(10):1397-1406, 1994.

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

[20] M. Sonka, V. Hlavac and R. Boyle, *Image processing, analysis and machine vision*, PWS publishing, San Francisco, 1999.

[21] M. Tuceyran and A.K. Jain, Texture analysis, in Handbook of Pattern Recognition and Computer Vision, C.H. Chen, L.F. Pau and P.S.P. Wang (Eds.), chapter 2, 235-276, World Scientific, Singapore, 1993.

[22] L. vanGool, P. Dewaele and A. Oosterlinck, Texture analysis, *Computer Vision, Graphics and Image Processing*, 29:336-357, 1985.

[23] Vistex Database
http://www-white.media.mit.edu/vismod/imagery/VisionTexture/vistex.html

[24] 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*,pp.269-285,1976.

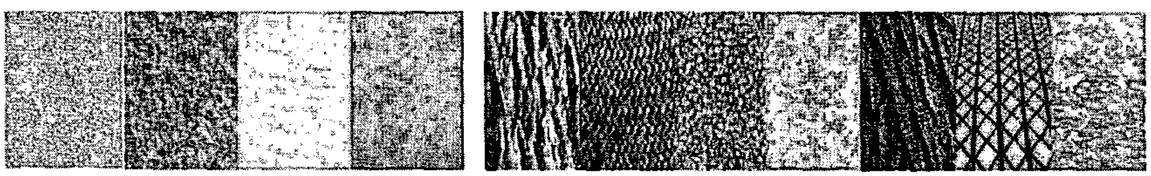

Fig.1 (a) Samples of Meastex data (b) Samples of Vistex data

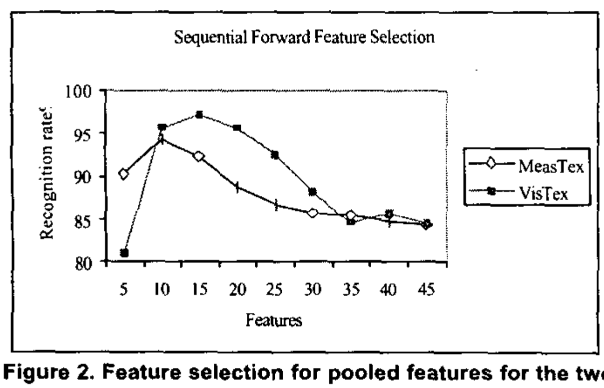

Figure 2. Feature selection for pooled features for the two