Image Retrieval using Multiscalar Texture Co-occurrence Matrix

S. K. Saha[1], A. K. Das[2] and Bhabatosh Chanda[3]

[1] CSE Department, Jadavpur University, Kolkata 700 032, India

[2] CST Department, Bengal Engineering & Sc. University, Howrah 711 103, India

[3] ECSU, Indian Statistical Institute, Kolkata 700 108, India

Abstract. We have designed and implemented a texture based image retrieval system that uses multiscalar texture co-occurrence matrix. The pixel array corresponding to an image is divided into a number of blocks of size 2×2 and a scheme is proposed to compute texture value for each of these blocks and then the texture co-occurrence matrix is formed. Image texture features are determined based on this matrix. Finally, a multiscalar version of the method is presented to cope with the texture pattern of various scale. Experiment using Brodatz texture database shows that retrieval performance of the proposed features is better than that of gray-level co-occurrence matrix and wavelet based features.

1 Introduction

Texture is a feature that has been extensively explored by various research group. Texture features are measured using either

(i) signal processing or statistical model [1], or

(ii) human perception model [2].

Texture is an innate property of virtually all object surfaces, including fabric, bark, water ripple, brick, skin, etc. In satellite image texture of a region can distinguish among grass land, beach, water body, urban area, etc. In [3] Haralick et al proposed the *co-occurrence matrix* representation of texture features. First joint occurrence of gray-level is represented as a matrix based on spatial distance and direction, and then meaningful statistics are extracted from the matrix. However, among those texture features *contrast*, *moments* and *entropy* are found to have greatest discriminatory power [4]. Smith and Chang [5, 7] extracted some statistics (e.g., mean and variance) from the Wavelet sub-bands and used them as texture features. To extract appropriate texture features Ma and Manjunath [8] explored various representation of Wavelet transforms, including orthogonal and bi-orthogonal Wavelet transforms, tree-structured Wavelet transforms, and Gabor Wavelet transforms. Bi-orthogonal Wavelet transforms is also used by Ko et al [9] in their work. Smith et al [6] and Berman et al [10] have dealt with Haar wavelet based texture representation. Gabor Filters are also used by Fournier et al [11].plan at. al [12] used fractal dimension as a measure of texture property. Tamura at. al [2]

K. Saha S., K. Das A. and Chanda B. (2006).
Image Retrieval using Multiscalar Texture Co-occurrence Matrix.
In *6th International Workshop on Pattern Recognition in Information Systems*, pages 136-145
DOI: 10.5220/0002477401360145

viewed texture features from a different angle and used six visually meaningful texture properties, including *coarseness*, *contrast*, *directionality*, *line-likeness*, *regularity*, and *roughness*. Sharing a similar view Liu and Picard [13] used *periodicity*, *directionality* and *randomness* as texture features. Kelly et al [14] has described the texture in terms of skewness and kurtosis. Wold decomposition is tried by Pentland and Picard [15]. A steerable pyramid based representation has been proposed by Aggarwal et al [16]. Gaussian Markovian random field based representation [17] is also used for denoting texture. Proposals of texture description using Luminance gradient [18], and Y values of YIQ model [19] are also made by the researchers. Ciocca et al [20] has described texture on the basis of neighbourhood gray tone matrix.

The early work reveals that among the classical approaches the co-occurrence matrix [3] based feature description is widely used. As the joint probability of gray-level of a predefined pair of pixels indicates, in some sense, the pattern of intensity variation, the gray-level co-occurrence matrix is used for texture measure. Based on the matrix features like energy, entropy, contrast, homogenity and moments are computed.

The current trend is towards the description of texture based on wavelet transformation. In [1], Manjunath and Ma have suggested a scheme which is as follows. In first level, the image is decomposed into four sub images of which one is low pass subimage and remaining three are orientation selective high pass subimages. In subsequent iteration, only the low pass subimage is decomposed into four subimages. Thus, after n iterations, we obtain n low pass subimages and $3 \times n$ high pass directional subimages. Corresponding to each subimage, energy, E_k (average of absolute intensity) and standard deviation of intensity values, σ_k are computed and used as features.

In this work, we present multiscalar texture co-occurrence matrix as the texture descriptor and the performance is compared with that of gray-level co-occurrence matrix and the wavelet based features described earlier. Section 2 describes the proposed features. Experimental results and comparison are presented in section 3 and concluding remarks are given in section 4.

2 Proposed Texture Features

We perceive the texture as repetitive or quasi-repetitive patterns. However, the gray-level co-occurrence matrix [3] captures only the co-occurrence of the intensity values, it can not reflect the repetitive nature of the patterns in true sense. In order to overcome this limitation, we propose texture co-occurrence matrix for describing the texture of the image.

2.1 Texture Co-occurrence Matrix

Usually a small patch of finite area of an image is required to feel or measure local texture value. The smallest region for such purpose could be a 2×2 block. So in order to compute the texture co-occurrence matrix, the intensity image is divided into blocks of size 2×2 as represented in Fig. 1(a).

Then gray level of the block is converted to binary [see Fig. 1(b)] by thresholding at the average intensity. Thus, the conversion from 2×2 intensity block to binary block

I_{00}	I_{01}
I_{10}	I_{11}

(a)

b_{00}	b_{01}
b_{10}	b_{11}

(b)

Fig. 1. (a) 2×2 intensity block and (b) corresponding Binary block [see text].

is done as follows:

$$b_{ij} = \begin{cases} 0 & \text{if } I_{ij} \leq t \\ 1 & \text{if } i_{ij} > t \end{cases}$$

where $t = \bar{I}$. This operation is same as the method of obtaining binary pattern in case of block truncation coding [21]. The 2×2 binary pattern obtained this way provides an idea of local texture within the block. By arranging this pattern in raster order a binary string is formed and corresponding decimal equivalent is its texture value. Thus,

$$\text{Texture value, } t_i = b_{00} \times 2^3 + b_{01} \times 2^2 + b_{10} \times 2^1 + b_{11} \times 2^0$$

Some examples of the blocks and corresponding texture values are shown in Fig. 2. Thus we get 15 such texture value as block of all 1's does not occur.

	(a)	(b)	(c)	(d)
Intensity Block	20 22 / 8 7	9 19 / 11 20	17 11 / 18 9	20 7 / 6 23
Binary Pattern	1 1 / 0 0	0 1 / 0 1	1 0 / 1 0	1 0 / 0 1
Binary Code	1100	0101	1010	1001
Texture value	12	5	10	9

Fig. 2. Example of intensity blocks and texture values.

A problem of this approach is that a smooth intensity block [see Fig. 3(b)] and a coarse textured block [see Fig. 3(a)] may produce same binary pattern and hence, same texture value. To surmount this problem we define a smooth block as having intensity variance less than a small threshold. In our experiment, threshold is 0.01 of average intensity variance computed over all the blocks. All such smooth blocks have texture value 0 as shown in Fig. 3(c).

Texture values of the blocks that we have obtained represent the texture pattern quite effectively. But, it suffers from one major drawback. As it has been shown in Fig. 4, two blocks varying widely in terms of their local contrast can have similar binary pattern resulting into same texture value. Thus, the texture value represents only the texture pattern and does not retain the contrast information.

Intensity Block	20	7	8	7	8	7
	6	23	6	8	6	8
Binary Pattern	1	0	1	0	0	0
	0	1	0	1	0	0
Binary Code	1001		1001		0000	
Texture value	9		9		0	
	(a)		(b)		(c)	

Fig. 3. Coarse and smooth blocks.

Intensiy Block	20	22	230	190
	8	7	10	20
Binary Pattern	1	1	1	1
	0	0	0	0

Fig. 4. Blocks with same texture values but varying in contrast.

In order to get rid of such problems, the representation of texture value of a block needs modification. So that, contrast information can be incorporated. The correcting factor is to be introduced in such a way so that

– In the corrected texture value, contrast information is reflected.
– Contribution of contrast information must not dominate over that of texture pattern of the block.
– Two blocks with different texture pattern must have different texture value even after the inclusion of correcting factor.

Keeping all the factors in mind, T_i, new texture value of a block is computed as follows:

$$T_i = t_i \times 16 + \frac{\delta_i}{16}$$

where, δ_i is the contrast in the 2×2 block. In our experiment,

$$\delta_i = g_{i1} - g_{i0}$$

where, g_{i1} is the average value of the pixels in the block marked as 1 and g_{i0} is the same for the pixels marked as 0. Thus, $g_{i1} - g_{i0}$ represents the contrast of the block. As the old value is multiplied and contrast is divided by 16, it is ensured that contrast information will not dominate over the contribution of texture pattern and blocks with different texture pattern will have different values.

(a) (b)

Fig. 5. (a) An image, (b) Corresponding texture image.

Thus, we get the scaled image whose height and width are half of that of the original image and the pixel values are same as texture values. This new image may be considered as the image representing the texture of the original image [see Fig. 5]. Once the texture image is obtained, co-occurrence matrix corresponding to that is computed following the methodology described in [3].

The texture measures must be independent of translation, rotation and size. To make the texture matrix translation invariant the 2×2 block frames are shifted by one pixel horizontally, vertically and both. For each case, co-occurence matrix is computed. To make the measure flip invariant, co-occurence matrices are also computed for the mirrored image. Thus, we have sixteen such matrices. Then, we take the element-wise average of all the matrices and normalize them to obtain the final one. Size and rotation is taken care of by the definition of co-occurrence matrix. In case of landscape, this is computed over the whole image; while in case of image containing dominant object(s) the texture feature is computed over the segmented region(s) of interest only.

The texture co-occurrence matrix provides the detailed description of the image texture, but handling of such multivalued feature is always difficult, particularly in the context of indexing and comparison cost. Hence, to obtain more perceivable features, based on the matrix the statistical measures like energy, entropy, variation in texture, Homogeneity in texture, texture moments are computed as it is done in case of gray-level co-occurrence matrix. We have considered moments of order 1 and 2 as the higher orders are not perceivable. Thus, a 9-dimensional feature vector is obtained to represent the texture of an image.

2.2 Multiscalar Texture Co-occurrence Matrix

Texture of an image can be classified as micro or macro texture. It depends of the size of pattern that repeats in the image. Thus, a good texture measure should have the capability to handle the patterns of various scale. The proposed feature, the way it has been described so far, is capable of handling the micro textures. In order to enhance its power to represent the macro texture we further improve it and propose multiscalar texture co-occurrence matrix.

It is computed as follows. In first iteration, we compute the features correspond to the original gray scale image. From it the texture co-occurrence matrix is generated and features are computed. In subsequent iteration, we consider only the pixels of alternate row and column in the image used for computation of feature in previous iteration [see Fig. 6]. It is similar to scaling function of lazy wavelet transform [22]. Texture

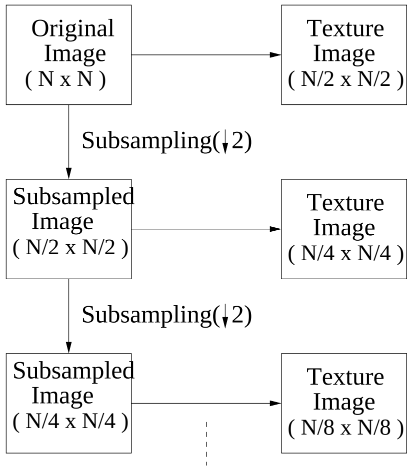

Fig. 6. Computation of multiscalar texture matrix.

co-occurence matrix and subsequently the features are computed corresponding to this new image whose height and size are reduced to half of those in the image used in previous iteration. Thus, in subsequent iteration when 2×2 blocks are formed, along with a particular pixel a new set of pixels in both horizontal and vertical directions are considered. As if, the scale of the texture pattern is extended in both the directions. In our experiment, we have repeated the process up to 3 iterations and a 27-dimensional feature vector is obtained.

3 Experimental Results

In order to evaluate the performance of the proposed texture features, we have carried out an experiment using the images of Brodatz texture database[4]. The database contains 112 images of size 640×640. Each image is divided into 25 sub-images of size 128×128. Thus, the final database contains 2800 images of 112 different classification and each classification has 25 images. Each database image is used as the query image and top order images are retrieved using Euclidean distance based exhaustive search. Precision of retrieval, $P(N)$ is measured after retrieving top N matches. In our experiment, the values of N are taken as 10, 20 and 25.

To study the performance of proposed feature, at first multiscalar version is not considered. Still the retrieval performance of proposed texture co-occurrence matrix based features is better than that of gray-level co-occurrence matrix based features as it has been shown in Table 1. Finally, the multiscalar version of the proposed matrix is used. Table 2 clearly shows that it outperforms the non-multiscalar version.

A few sample retrieval results based on proposed multiscalar texture features are shown in Fig. 7. In order to compare the performance of proposed texture measures, we

[4] *www.ux.his.no/∼tranden/brodatz.html*

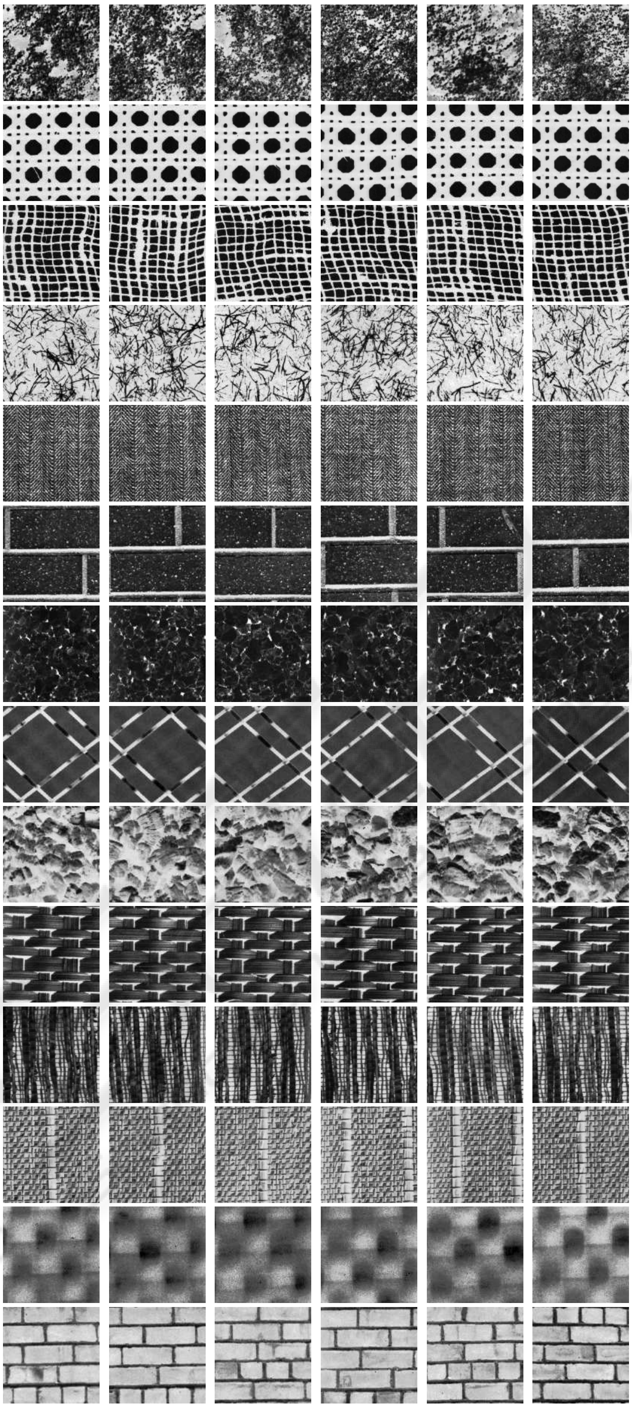

Fig. 7. Retrieval results using Brodatz database: showing top 5 matches excluding the query image itself; first image of each row is the query image.

Table 1. Comparison of precision (in %) using Brodatz database.

	Gray-level Co-occurence Matrix	Texture Co-occurence Matrix
P(10)	56.33	60.03
P(20)	44.29	48.46
P(25)	39.84	44.25

Table 2. Comparison of precision (in %) using Brodatz database.

	Texture Co-occurrence Matrix	Multiscalar Texture Co-occurrence Matrix
P(10)	60.03	74.31
P(20)	48.46	62.23
P(25)	44.25	56.48

have computed the multiscalar gray-level co-occurrence matrix based features following the similar technique to obtain 27-dimensional feature vector after three iterations. We have also implemented a system to work with wavelet based feature [1] as discussed in section 1. We have decomposed the image in 3 iteration. Thus, 12 different subimages are obtained and finally, a 24-dimensional feature vector is computed. The performance of these three types of features are shown in Table 3 and Fig. 8. It is clear that precision of retrieval using the proposed measure is better than others.

Table 3. Comparison of precision (in %) using Brodatz database.

	Multiscalar Texture Co-occurrence Matrix	Multiscalar Gray-level Co-occurrence Matrix	Wavelet
P(10)	74.31	68.17	72.89
P(20)	62.23	54.89	60.18
P(25)	56.48	49.47	54.16

4 Conclusion

In this work we have presented the idea of computing local texture based on blocks of 2×2 pixels of the intensity image. Each block is assigned a texture value and a texture image corresponding to the original image is obtained. The co-occurrence matrix corresponding to this texture image provides detailed description of the image texture. As the

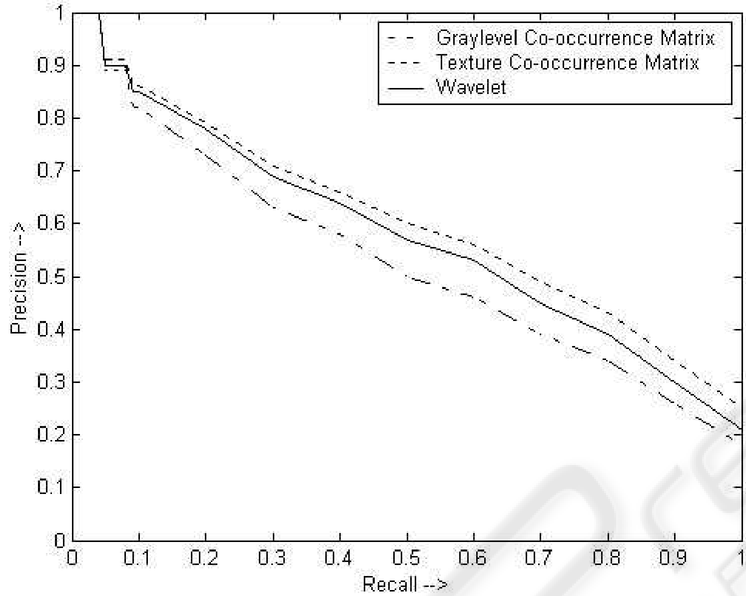

Fig. 8. Recall-Precision graph using various texture features for Brodatz database.

proposed matrix reflects the co-occurrence of the texture of 2×2 blocks, it has stronger capability than the gray-level co-occurrence matrix for representing the texture. As the texture co-occurrence matrix is high dimensional and detailed, more perceivable features, like entropy, energy, texture moments, homogeneity and variation in texture are computed from the matrix. Finally, multiscalar texture measure is proposed. So that, proposed scheme can handle various micro or macro textures. Experiment shows the performance of proposed measure is better than that of gray-level co-occurrence matrix and wavelet based measures.

References

1. Manjunath, B. S., Ma, W. Y.: Texture features for browsing and retrieval of image data. IEEE Trans. on PAMI Vol. 18 (1996) 837–842
2. Tamura, H., Mori, S., Yamawaki, T.: Texture features corresponding to visual perceptoin. IEEE Trans. on SMC Vol. 8(6) (1978) 460–473
3. Haralick, R. M.,Shanmugam, K., Dinstein, I.: features for image classification. Trans. on SMC Vol. 3(11) (1973) 610–622
4. Gotlieb, C. C., Kreyszig H. E.: Texture descriptor based on co-occurrence matrices. Computer Vision, Graphics and Image Processing Vol. 51 (1990) 70–86
5. Smith, J. R., Chang, S. F.: Transform features for texture classification and discrimination in large image databases. In: Proc. of IEEE Intl. Conf. on Image Processing Vol. 3 (1994) 407–411

6. Smith, J. R., Chang, S. F.: Automated image retrieval using color and texture. Technical Report CU/CTR 408-95-14, Columbia University (1995)
7. Smith, J. R., Chang, S. F.: Automated binary texture feature sets for image retrieval. In: Proc. of IEEE Intl. Conf. on ASSP, USA (1996) 2239–2242
8. Ma, W. Y., Manjunath, B. S.: A comparison of wavelet transform features for texture image annotation. In: Proc. of IEEE Intl. Conf. on Image Processing (1995) 256–259
9. Ko, B., Peng, J., Byun, H.: Region-based image retrieval using probabilistic feature relevance learning. Pattern Analysis and Applications Vol. 4 (2001) 174–184
10. Berman, A. P.,Shapiro, L. G.: A Flexible Image Database System for Content-Based Retrieval. Computer Vision and Image Understanding Vol. 75 (1999) 175–195
11. Fournier, J., Cord, M., Philipp-Foliguet, S.: Retin: A content-based image indexing and retrieval system. Pattern Analysis and Applications Vol. 4 (2001) 153–173
12. Kaplan, L. M.: Fast texture database retrieval using extended fractal features. SPIE Vol. 3312 (1998) 162–173
13. Liu, F., Picard, R. W.: Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. IEEE Trans. on PAMI Vol. 18(7) (1996) 722–733
14. Kelly, P. M., Cannon, T. M., Hush, D. R.: Query by Image Example: the CANDID approach. SPIE Vol. 2420 (1995) 238–248
15. Pentland, A., Picard, R.: Introduction to Special Section on the Digital Libraries: Representation and Retrieval. IEEE Trans. on PAMI Vol. 18 (1996) 769–770
16. Aggarwal, G., Dubey, P., Ghosal, S., Kulshreshtha, A., Sarkar, A.: ipure: Perceptual and user-friendly retrieval of images. In: Proc. of IEEE Conf. on Multimedia and Exposition, Vol. 2, New york, USA (2000) 693–696
17. Sciascio, E. D., Mingolla, G., Mongiello, M.: Content-based image retrieval over the web using query by sketch and relevance feedback. In: Proc. of the Third International Conf. VISUAL '99, Amsterdam (1999) 123–130
18. Brunelli, R., Mich, O.: Image retrieval by examples. IEEE Trans. on Multimedia Vol. 2(3) (2000) 164–171
19. Laaksonen, J., Koskela, M., Laakso, S., Oja, E.: Picsom - content-based image retrieval with self-organizing maps. Pattern Recognition Letters Vol. 21 (2000) 1199–1207
20. Ciocca, G., Gagliardi, I., Schettini, R.: Quicklook2: An Integrated Multimedia System. International Journal of Visual Languages and Computing Vol. 12 (2001) 81–103
21. Delp, E. J., Mitchell, O. R.: Image compression using block truncation coding. IEEE Trans. on Comm. Vol. 27 (1979) 1335–1342
22. Vetterli, M., Kovacevic, J.: Wavelets and Subband Coding. Prentice Hall, Englewood Cliffs, New Jersey, (1995)