A Rotation Invariant Pattern Operator for Texture Characterization

R Suguna1[†] and P. Anandhakumar 2[††],

[†] Research Scholar, Dept of IT, Madras Institute of Technology, Anna University, Chennai 600044 , Tamil Nadu, India.
[††]Assistant Professor, Dept of IT, Madras Institute of Technology, Anna University, Chennai 600 044 , Tamil Nadu, India.

Summary
This paper proposes a novel Ternary Pattern Operator for texture characterization. The devised operator has rotation-invariant capability. The operator is used for extracting texture features from the images. From these features the texture present in the image are identified. The operator can be realized with few operations in local neighborhood and hence it is computationally simple. In the training phase the classifier is trained with samples with one particular angle and tested with samples of different angles. A detailed analysis is presented for rotation angles of misclassified samples. A distribution based classification approach is used for discriminating the textures. A probabilistic model is built for each texture class. A classification problem of different Brodatz texture and seven rotation angles is used in experiments. Experimental results prove that the performance of the proposed operator for feature extraction is appreciable.

Key words:
Local Ternary Pattern Operator, Texture Characterization, Non parametric classification, Brodatz texture.

1. Introduction

In computer vision, the visual appearance of the view is captured with digital imaging and stored as image pixels. Texture analyst state that there is significant variation in intensity levels or colors between nearby pixels, and at the limit of resolution there is non-homogeneity. Spatial non-homogeneity of pixels corresponds to the visual texture of the imaged material which may result from physical surface properties such as roughness, for example. Image resolution is important in texture perception, and low-resolution images contain typically very homogenous textures. Also we all know that real-world objects and surfaces are not flat, nor uniform, and there are numerous potential computer vision applications that could utilize texture information.
Texture is a fundamental property of surfaces. It can be seen almost anywhere. For example, in outdoor scene images, trees, bushes, grass, sky, lakes, roads, buildings etc. appear as different types of texture. Hence, texture can be used as a measure of interpreting the images. Texture can be regarded as the visual appearance of a surface or material.
Typically textures and the analysis methods related to them are divided into two main categories with different computational approaches: the stochastic and the structural methods. Structural textures are often man-made with a very regular appearance consisting, for example, of line or square primitive patterns that are systematically located on the surface (e.g. brick walls). In structural texture analysis the properties and the appearance of the textures are described with different rules that specify what kind of primitive elements there are in the surface and how they are located. Stochastic textures are usually natural and consist of randomly distributed texture elements, which again can be, for example, lines or curves (e.g. tree bark). The analysis of these kinds of textures is based on statistical properties of image pixels and regions. The above categorization of textures is not the only possible one; there exist several others as well, for example, artificial vs. natural, or micro textures vs. macro textures. Regardless of the categorization, texture analysis methods try to describe the properties of the textures in a proper way. It depends on the applications what kind of properties should be sought from the textures under inspection and how to do that. This is rarely an easy task.
Texture is important in many image analysis and computer vision tasks. It gives additional information compared only to color or shape measurements of the objects. Sometimes it is not even possible to obtain color information at all, as in night vision with infrared cameras. Color measurements are usually more sensitive to varying illumination conditions than texture, making them harder to use in demanding environments like outdoor conditions. Therefore texture measures can be very useful in many real-world applications, including, for example, outdoor scene image analysis.
The major problem of using texture information is that textures have usually great variations in their visual appearance. Textures can be oriented and scaled differently and imaged under changing illumination conditions. These cause huge variation to the texture appearance and there are strict requirements for the texture

Manuscript received April 5, 2010
Manuscript revised April 20, 2010

measures to produce a reasonable description of the analyzed surfaces. The features should tolerate illumination variations and they should be able to handle different rotations and scales as well. Also computational issues are important in applications; for example, to achieve a reasonable classification time for textured samples.

To exploit texture in applications, the measures should be accurate in detecting different texture structures, but still be invariant or robust with varying conditions that affect the texture appearance. Computational complexity should not be too high to preserve realistic use of the methods. Different applications set various requirements on the texture analysis methods, and usually selection of measures is done with respect to the specific application.

The ultimate goal of texture characterization systems is to recognize different textures. Applications have different requirements for recognition: usually accuracy is the most important property, but sometimes also speed, usability and configurability should be prioritized. There is no universal recognition method for different texture characterization tasks. In industrial inspection applications, there is typically a trade-off between accuracy and speed. It is however almost equally important that the recognition system is easy to use and configure in new environments. In more general image analysis tasks, texture recognition methods must detect different textures in the images, but also consider images on a higher level. For example, in scene image analysis, the relations between different regions set constraints on the recognition and help the overall image understanding.

Most approaches to texture classification assume, either explicitly or implicitly, that the unknown samples to be classified are identical to the training samples with respect to spatial scale, orientation, and gray-scale properties. However, real-world textures can occur at arbitrary spatial resolutions and rotations and they may be subjected to varying illumination conditions. This has inspired a collection of studies which generally incorporate invariance with respect to one or at most two of the properties spatial scale, orientation, and gray scale.

There are many applications for texture analysis in which rotation-invariance is important, but many of the existing texture features are not invariant with respect to rotations

Early approaches proposed for rotation invariant texture classification include the methods based on texture anisotropy[1], polarograms [2] and generalized co-occurrence matrices [3].A method based on the circular symmetric autoregressive random field (CSAR) model for rotation-invariant texture classification is developed [4]. A method for classification of rotated and scaled textures using Gaussian Markov random field models was introduced by Cohen et al.[5]. Approaches based on Gabor filtering have been proposed by, among others, Leung and Peterson [6], Porat and Zeevi [7], and Haley and Manjunath [8]. A steerable oriented pyramid was used to extract rotation invariant features by Greenspan et al.[9] and a covariance-based representation to transform neighborhood about each pixel into a set of invariant descriptors was proposed by Madiraju and Liu [10].An extension of Laws' masks for rotation-invariant texture characterization is proposed [11]. Many papers have been published on plain rotation invariance analysis [12], [13], [14]. A number of techniques incorporating invariance with respect to both spatial scale and rotation have been developed [15], [16], [17]. Chen and Kundu [18] and Wu and Wei [19] approached gray scale invariance by assuming that the gray scale transformation is a linear function. Chen and Kundu realized gray scale invariance by global normalization of the input image using histogram equalization.

Characterization of textured materials is usually very difficult and the goal of characterization depends on the application. In general, the aim is to give a description of analyzed material, which can be, for example, the classification result for a finite number of classes or visual exposition of the surfaces.

In realistic texture characterization problems the amount of data is usually huge and sometimes there is no prior knowledge of the data available. Characterization should then be made in an unsupervised manner. In these cases, it might be very problematic to create representative models for the analyzed textures or study how well specific texture features perform.

The paper is organized as follows: Section 2 discusses various texture measures used for texture classification. Section 3 explains the proposed Operator. Section 4 briefs the classification approach. Experiments and Results are analyzed in Section 5. The conclusion of the method is in Section 6.

2. Texture Measures

Since we are interested in interpretation of images we can define texture as the characteristic variation in intensity of a region of an image which should allow us to recognize and describe it and outline its boundaries. The degrees of randomness and of regularity will be the key measure when characterizing a texture. In texture analysis the similar textural elements that are replicated over a region of the image are called texels. It is quite clear that a texture is a complicated entity to measure. The reason is primarily that many parameters are likely to be required to characterize it. The gray-level co-occurrence matrix approach is based on studies of the statistics of pixel intensity distributions. The early paper by Haralick et al.[20] presented 14 texture measures and these were used successfully for classification of many types of materials for example, wood, corn, grass and water. However, Conners and Harlow [21] found that only five of these measures were normally used, viz. “energy”, “entropy”, “correlation”, “local homogeneity”, and “inertia”. The size of the co-occurrence matrix is high and suitable choice of d (distance) and θ (angle) has to be made.

Recent developments include the work with automated visual inspection in work. Ojala et al., [22] and Manthalkar et al., aimed at rotation invariant texture classification. Pun and Lee [23] aims at scale invariance.

Ojala T & Pietikäinen M [24] proposed a multichannel approach to texture description by approximating joint occurrences of multiple features with marginal distributions, as 1-D histograms, and combining similarity scores for 1-D histograms into an aggregate similarity score.

Ojala T introduced a generalized approach to the gray scale and rotation invariant texture classification method based on local binary patterns [25]. The current status of a new initiative aimed at developing a versatile framework and image database for empirical evaluation of texture analysis algorithms is presented by him. A multiresolution approach to gray-scale and rotation invariant texture classification based on local binary patterns is presented [26].

Ahonen T, proved that the local binary pattern operator can be seen as a filter operator based on local derivative filters at different orientations and a special vector quantization function [27]. A rotation invariant extension to the blur insensitive local phase quantization texture descriptor is presented by Ojansivu V [28]. Block-based texture methods are proposed for content-based image retrieval by Takala V [29].

3. Local Ternary Pattern

The local binary pattern (LBP) operator is a theoretically simple, yet very powerful and gray-scale invariant method of analyzing textures (Ojala et al. 1996) [19]. In practice, the LBP operator combines characteristics of statistical and structural texture analysis: it describes the texture with micro-primitives, often called textons, and their statistical placement rules (Maenpaa & Pietikainen 2005) [20].

LBP histograms with multiple neighborhood parameters are created and concatenated. Large neighborhood radii result in sparse sampling. Feature vector grows linearly with the number of different neighborhoods. Also LBP is a crisp operator.

The original 8-bit version of the LBP operator considers only the eight nearest neighbors of each pixel and it is rotation variant, but invariant to monotonic changes in gray-scale. The dimensionality of the LBP feature distribution can be calculated according to the number of neighbors used. The basic 8-bit LBP can represent 2^8 different local patterns, so the dimensionality of the feature vector is 256. The definition of the LBP has been extended to arbitrary circular neighborhoods of the pixel to achieve multi-scale analysis and rotation invariance (Ojala et al. 2002).

In the multiresolution model of the LBP, separate operators at different scales are first constructed and the final feature vector is a combination of individual feature vectors created simply by concatenating them one after another. Combining different operators can also be done by constructing the joint distribution of all different LBP codes, but such a distribution would be very sparse and too large in realistic cases

The number of possible local patterns increases rapidly when the number of neighbor samples grows. For example with 16 neighbors, the size of the histogram would be 2^{16} bins, which is impractical to use in any realistic applications. Maenpaa et al. [20] suggested considering only the so-called ’ uniform ’ patterns, where the maximum number of bit-wise changes from one to zero or vice versa in the circular neighborhood is limited. Usually the maximum number of bit changes is allowed to be two. It was observed that certain local patterns seem to represent the great majority, sometimes over 90 percent of all local patterns in the image. With this approach, the number of different binary codes is reduced dramatically, but the discrimination performance remains good. For example, the number of the histogram bins with an 8-bit LBP is reduced to 59 bins, where 58 are the actual uniform patterns and the last one contains all the others. Rotation invariance of LBP is achieved considering only a small rotation invariant subset of the original binary patterns [20]. They are constructed rotating the obtained binary pattern clockwise so many times that the maximum number of the most significant bits is zero. Practically, the rotation invariance is achieved in terms of normalization of the binary code, and can be easily implemented with look-up tables. Ojala et al. (2002b) extended this approach to multi-scale for making the LBP operator more suitable for arbitrary rotation angles. They calculated the rotation invariant operator from different scales using ’ uniform’ patterns to keep the size of the feature distribution reasonable.

Our approach is an extension of LBP operator but examines the similarity among the pixels. Pixels with slight variations are considered alike and the degree of similarity is judged with threshold value.

Local Ternary Pattern (LTP) considers 8 nearest neighbors of each pixel and uses concept of “uniform” patterns. But the number of histogram bins is reduced to 24 almost half of that of LBP approach. With this reduced dimension, the operator performs well.

3.1 Gray scale and Rotation Invariant LTP

Let us define texture T as the joint distribution of the gray levels of P+1 (P>0) image pixels:

$$T = t(g_c, g_0, g_1, \ldots, g_{P-1}), \quad (1)$$

where g_c corresponds to the gray value of the center pixel of a local neighborhood. g_p(p=0,1,2,…,P-1) correspond to the gray values of P where P=8.

Without losing information, g_c can be subtracted from g_p :

$$T = t(g_c, g_0-g_c, g_1-g_c, \ldots, g_{P-1}-g_c), \quad (2)$$

Assuming that the differences are independent of g_c , the distribution can be factorized:

$$T = t(g_c)t(g_0-g_c, g_1-g_c, \ldots, g_{P-1}-g_c), \quad (3)$$

Since $t(g_c)$ describes the overall luminance of an image, which is unrelated to local image texture, it can be ignored:

$$T = t(g_0-g_c, g_1-g_c, \ldots, g_{P-1}-g_c) \quad (4)$$

The grey level difference between the pixels around 8 neighbors is checked against a threshold value θ.

$$T = t(y(g_0-g_c), y(g_1-g_c), \ldots, y(g_{P-1}-g_c)) \quad (5)$$

where

$$y(x) = \begin{cases} 0 & x < -\theta \\ 1 & -\theta \le x \le \theta \\ 2 & x > \theta \end{cases} \quad (6)$$

The gray-scale invariance is achieved by means of determining the y value by comparison instead of using their exact values. The y value will not be affected by shift in the gray values. Fig.1. shows the y values calculated along the border of a 3 x 3 local region and its Local Ternary Pattern. The Pattern String can be formed from the Pattern Unit matrix by collecting the y values starting from any position.

124	138	145
126	129	134
137	45	127

3x3 region Pattern Matrix

1	2	2
1		1
2	0	1

Pattern String : 22110211

Fig, 1 Generation of LTP String

Achieving Rotation Invariance

To remove the effect of rotation, each LTP code must be rotated back to a reference position, effectively making all rotated versions of a binary code the same.

To achieve this the type of transitions is examined and coded. Successive values of the pattern string are examined and the possible pairs are $0 \leftrightarrow 0$, $1 \leftrightarrow 1$, $2 \leftrightarrow 2$,$0 \leftrightarrow 1$, $0 \leftrightarrow 2$, $1 \leftrightarrow 2$. Among this if the successive pairs are of same value, then there is no transition. ($0 \leftrightarrow 0$, $1 \leftrightarrow 1$, $2 \leftrightarrow 2$). $0 \leftrightarrow 1$ indicates the successive pairs can be (0 1) or (1 0). The type of transitions are coded.

This transformation can be defined as follows:

$$FT = \{PS(i, i+1) | i = 0, 1, \ldots, P-1\} \quad (7)$$

where PS is the Pattern String and PS(x,y) denotes the successive pair and is defined as follows.

$$PS(x,y) = \begin{cases} 0 & x = y \\ 1 & (x,y) = 0 \leftrightarrow 1 \\ 2 & (x,y) = 0 \leftrightarrow 2 \\ 3 & (x,y) = 1 \leftrightarrow 2 \end{cases} \quad (8)$$

FT indicates findTransition operator and contains the recorded transition types. Table 1 shows the generated FT for the given Pattern String.

Pattern String	FT
1 1 2 2 2 0 0 0	0 3 0 0 2 0 0 1
2 2 2 1 1 1 1 1	0 0 3 0 0 0 0 3
0 0 2 1 0 1 1 1	0 2 3 1 1 0 0 1
0 0 2 2 2 0 0 0	0 2 0 0 2 0 0 0

Table1 Transition Coding

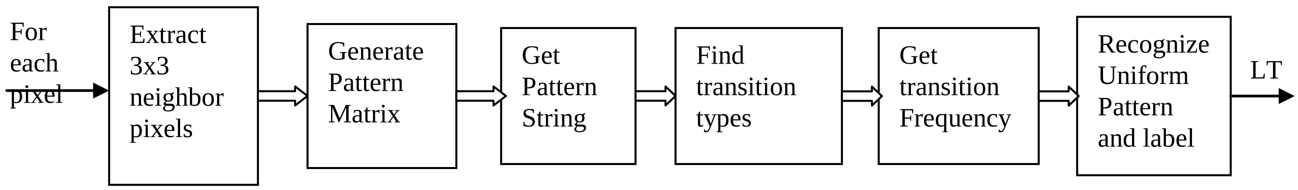


Fig 2 Generation of LTP

The frequency of transitions $fnum$ is calculated from FT.

$$fnum = [n1\ n2\ n3] \qquad (9)$$

where $n1, n2, n3$ are number of $0 \leftrightarrow 1$, $0 \leftrightarrow 2$, $1 \leftrightarrow 2$ transitions respectively.

The concept of "uniform" patterns was introduced since certain patterns seem to be fundamental properties of texture, providing the vast majority of patterns, sometimes over 90%. These patterns are called "uniform" because they have one thing in common: at most two transitions occur in FT.
Formally we define the uniformity measure U as:

$$U = \left(\sum fnum \ \leq 2\right) \qquad (10)$$

The uniform patterns are extracted and for each pattern, a texture unit number is assigned. All other patterns are labeled "non uniform " and collapsed into one value as 24. The proposed representation has 23 uniform patterns and few of them with their corresponding texture unit number is listed in Table 2

Uniform Pattern	Texture Unit Number
00000000	0
00000001	1
00000011	2
00001111	4
01111111	7
11111112	9
11112222	12
11222222	14
12222222	15
22222222	16
00000002	17
00000222	19
00002222	20
00222222	22

Table 2 : Texture unit number

The occurrence frequency of LTP over the larger region of an image will reveal the textural characteristics of the image and it will be used as a global image descriptor.
The steps involved in the generation of LTP is depicted in Fig. 2.

4. LTP for classification

We have proposed a simple and robust rotation invariant operator. In our work, we have used 3x3 neighborhood to capture micro-features. For a given block of image, the 8 neighborhood is considered to obtain the LTP value. We have reduced the number of bins of histogram.

Our proposed approach has two phases :

- Generation of Model for Concept
- Testing phase.

The steps involved in generation of Model for Class is illustrated in Fig. 3. Testing Phase uses a non-parametric classification principle.

4.1 Non-parametric Classification Principle

In classification, the dissimilarity between a sample and a model LBP distribution is measured with a non-parametric statistical test. This approach has the advantage that no assumptions about the feature distributions need to be made. Sokal & Rohlf (1969) have called this measure the *G* statistic:

$$G(S, M) = 2\sum_{b=1}^{B} S_b \log \frac{S_b}{M_b} = 2\sum_{b=1}^{B} [S_b \log S_b - S_b \log M_b] \qquad (11)$$

where *S* and *M* denote (discrete) sample and model distributions, respectively. S_b and M_b correspond to the probability of bin *b* in the sample and model distributions. *B* is the number of bins in the distributions.

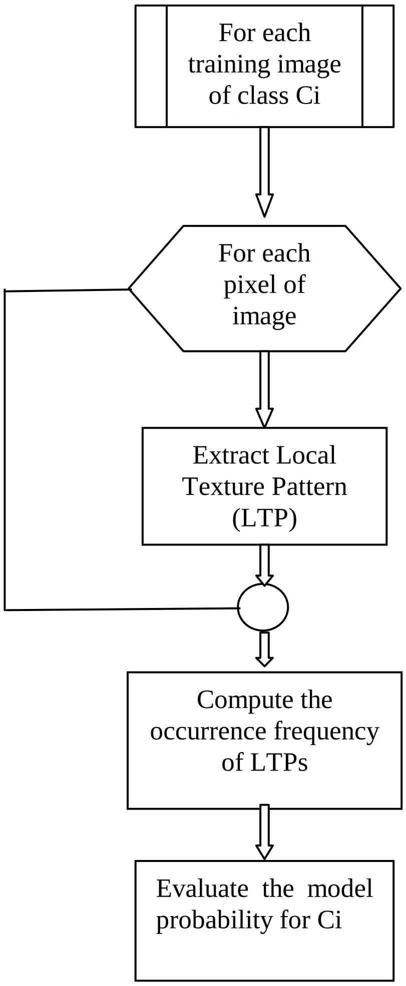


Fig. 3 Steps in Generation of Class Model

The *G* statistic can be used in classification in a modified form:

$$L(S,M) = -\sum_{b=1}^{B} S_b \log M_b \quad (12)$$

Model textures can be treated as random processes whose properties are captured by their LBP distributions. In a simple classification setting, each class is represented with a single model distribution *i M* . Similarly, an unidentified sample texture can be described by the distribution *S* . *L* is a pseudo-metric that measures the likelihood that the sample *S* is from class *i* . The most likely class *C* of an unknown sample can thus be described by a simple nearest-neighbor rule:

$$C = {}^{\arg\min}{}_{i} L(S, M^i) \quad (13)$$

5. Experiments and Results

We demonstrate the performance of our approach with the proposed LTP operator with texture image data that have been used in recent studies on rotation invariant texture classification. Since the training data included samples from several rotation angles, we also present results for a more challenging setup, where the samples of just one particular rotation angle are used for training the texture classifier, which is then tested with the samples of the other rotation angles.

5.1 Image Data and Experimental Setup

The image data included 13 textures from the Brodatz album. Textures are presented at 6 different rotation angles (0, 30, 60, 90, 120, 150). For each texture class there were 16 images for each class and angle (hence 1248 images in total) . Each texture class comprises following subsets of images: 16 'original' images, 16 images rotated at 30^0 , 16 images rotated at 60^{0,} 16 images rotated at 90^{0,} 16 images rotated at 120^{0}and16 images rotated at 150^0. The size of each image is 128x128. The texture classes considered for our study are shown in Fig. 4

5.2 Contribution of “uniform” patterns

We studied the contribution of “uniform patterns under different threshold values for each texture class. Table 3 lists the proportions of “uniform” patterns for the texture classes used in our study.
When threshold values are increased, the proportions (%) of “uniform” patterns improves, but only few “uniform” patterns exists ie., the distribution of “uniform” patterns is reduced.
It is also desired to estimate the optimum threshold for the texture classification. This evaluation can be done by studying the performance of the classification.

Threshold θ	Texture classes			
	Bark	Brick	Bubbles	Grass
5	28.77	33.83	30.32	41.7
10	38.05	61.66	41.9	43.2
20	71.11	89.05	79.53	45.41
30	88.93	96.65	92.12	64.90

Table 3 Contributions (%) of “uniform” patterns for each Texture class used in the experiment with varied threshold

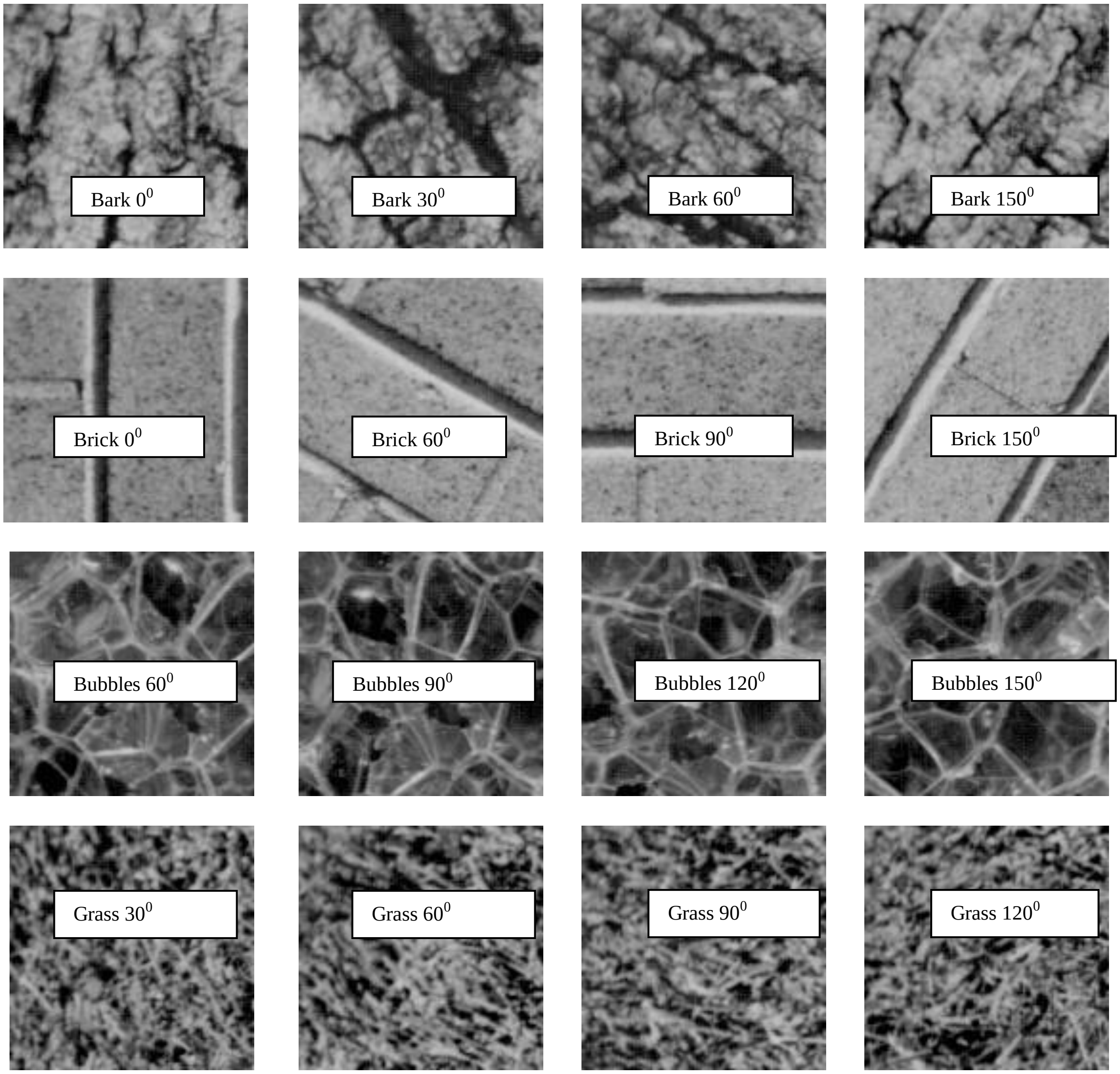


Fig 4 : Image Samples used for Classification

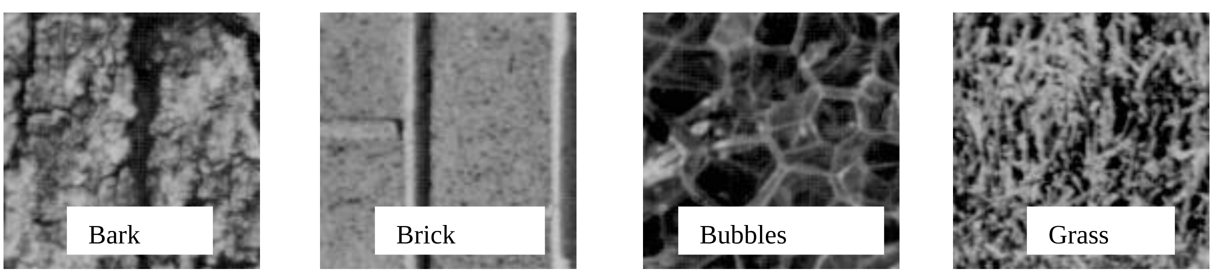
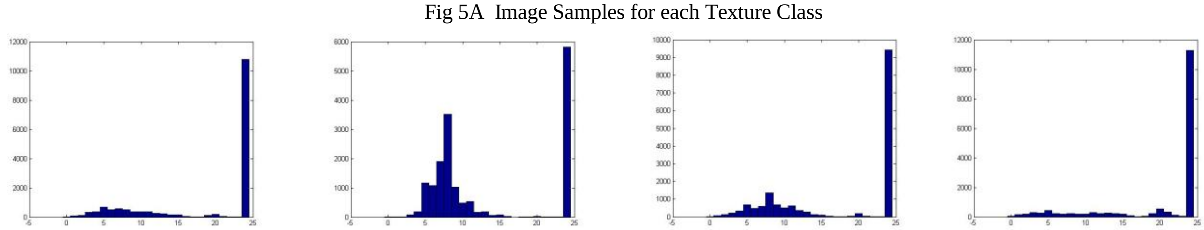


Fig 5A Image Samples for each Texture Class



Bark Brick Bubble Grass

Fig 5B Histogram of “uniform” patterns for each Texture Class

Distribution of patterns for each texture class is shown in Fig. 5A and Fig 5B.
Table 4 presents results for a the challenging experimental setup where the classifier is trained with samples of just one rotation angle and tested with samples of other rotation angles for various thresholds.

Texture	Classification Accuracy (%) for different Training angles				
	30^0	60^0	90^0	120^0	150^0
Bark	93.75	93.75	93.75	93.75	93.75
Brick	87.5	75.0	75.0	81.25	68.75
Bubbles	87.5	87.5	100.0	93.75	81.25
Grass	93.75	100.0	87.5	93.75	87.5

Table-4 Classification Accuracies (%) with classifier trained with one rotation angle(0^0) with threshold (θ=5) and Tested with other versions.

Texture	Classification Accuracy (%) for different Training angles				
	30^0	60^0	90^0	120^0	150^0
Bark	87.5	75.0	75.0	75.0	81.25
Brick	93.75	87.5	87.5	93.75	87.5
Bubbles	100.0	93.75	100.0	100.0	93.75
Grass	100.0	100.0	100.0	100.0	93.75

Table-5 Classification Accuracies (%) with classifier trained with one rotation angle(0^0) with threshold (θ=10) and Tested with other versions.

Texture	Classification Accuracy (%) for different Training angles				
	30^0	60^0	90^0	120^0	150^0
Bark	68.75	75.0	68.75	87.5	87.5
Brick	93.75	87.5	93.75	93.75	87.5
Bubbles	100.0	100.0	100.0	87.5	100.0
Grass	100.0	100.0	100.0	100.0	93.75

Table-6 Classification Accuracies (%) with classifier trained with one rotation angle(0^0) with threshold (θ=20) and Tested with other versions.

Texture	Classification Accuracy (%) for different Training angles				
	30^0	60^0	90^0	120^0	150^0
Bark	68.75	68.75	68.75	81.25	81.25
Brick	93.75	87.5	100.0	100.0	87.5
Bubbles	93.75	68.75	81.25	87.5	87.5
Grass	100.0	100.0	100.0	93.75	93.75

Table-7 Classification Accuracies (%) with classifier trained with one rotation angle(0^0) with threshold (θ=30) and Tested with other versions.

Results clearly state that classification varies with different values of threshold values. Fig . 6 show the overall classification accuracies in percentage for varied threshold values.

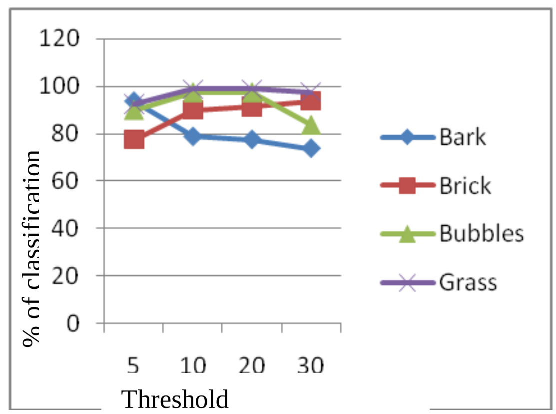


Fig. 6 Threshold Vs Classification Performance

Fig 7 shows the contribution of "uniform" patterns in each texture class for varied threshold values.

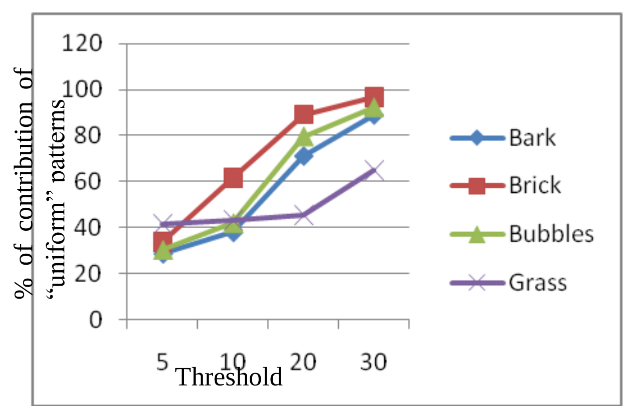


Fig. 7 Threshold Vs Proportion(%) of

Observations clearly state that the threshold plays a major role in classification. Finding an optimum threshold value for a given set of texture classes seems to be a tough task. This is due to variations in texture images. A non parametric classification principle is used for texture discrimination.

The overall recognition rate is compared with other texture measures such as co-occurrence matrix, autocorrelation method and laws texture measure. The performances of different texture measures are listed in Table 8.

Texture method	Recognition rate in %
Autocorrelation	76.1
Co occurrence	78.6
Laws	82.2
LTP	93.5

Table 8. Recognition rates of various Texture measures

6. Conclusion

We presented a theoretically and computationally simple yet efficient Local Ternary Operator. The operator can be used for gray-scale and rotation invariant texture classification. The approach is based on "uniform" patterns and nonparametric discrimination of sample and prototype distributions. "Uniform" patterns help us to recognize texture microstructures such as edges. Our operator handles similarity among the pixels and this is measured by varying the threshold values. This also includes idea of human perception. We developed a generalized gray-scale and rotation invariant operator LTP for eight neighbors. We combined the responses of texture class for multiple angles to estimate the performance of classification.

Experimental results are appreciable where the classifier is trained with original version of image samples and tested with different rotated versions. The proposed operator performs well for the structural and stochastic patterns.

Computational simplicity is another advantage since the operator is evaluated with few comparisons in a local neighborhood. This facilitates less time for implementation. The performance can be further enhanced by using different classifiers. The operator may be suitably improved to achieve scale invariance.

Acknowledgements

We wish to thank Dr. Kimmo Valkealahti from Nokia Research Center and Professor Erkki Oja from Helsinki University of Technology for providing these images. We have downloaded the images from Outex site.

References

[1] P. Chetverikov, "Experiments in the rotation-invariant texture discrimination using anisotropy features", in Proc. 6th International Conference on Pattern Recognition, Munich, Germany, 1071-1073 (1982).

[2] L.S. Davis, "Polarograms: a new tool for texture analysis", Pattern Recognition 13, 219-223 (1981).

[3] L.S. Davis, S.A. Johns and J.K. Aggarwal, "Texture analysis using generalized co-occurrence matrices", *IEEE Transactions on* Pattern Analysis and Machine Intelligence **1**, 251-259 (1979).

[4] R.L. Kashyap and A. Khotanzad, A model-based method for rotation invariant texture classification, *IEEE Transactions on Pattern* Analysis and Machine Intelligence **8**, 472-481 (1986).

[5] F.S. Cohen, Z. Fan and M.A. Patel, Classification of rotated and scale textured images using Gaussian Markov random field models, IEEE Transactions on Pattern Analysis and Machine Intelligence **13**, 192-202 (1991).

[6] M.M. Leung and A.M. Peterson, Scale and rotation invariant texture classification, in *Proc. 26th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA (1992).

[7] M. Porat and Y. Zeevi, Localized texture processing in vision: analysis and synthesis in the Gaborian space, *IEEE Transactions* on Biomedical Engineering **36**, 115-129 (1989).

[8] G.M. Haley and B.S. Manjunath, Rotation-invariant texture classification using modified Gabor filters, in Proc. IEEE Conference on Image Processing, Austin, TX, 655-659 (1994).

[9] H. Greenspan, S. Belongie, R. Goodman and P. Perona, Rotation invariant texture recognition using a steerable pyramid, in Proc. 12th International Conference on Pattern Recognition, vol. 2, Jerusalem, Israel, 162-167 (1994).

[10] S.V.R. Madiraju and C.C. Liu, Rotation invariant texture classification using covariance, in Proc. IEEE Conference on Image Processing, Vol. 1, Washington, D.C., 262-265 (1995).

[11] J. You and H.A. Cohen, Classification and segmentation of rotated and scaled textured images using texture "tuned" masks, Pattern Recognition 26, 245-258 (1993).

[12] Lam, W.-K., Li, C.-K.: Rotated Texture Classification by Improved Iterative Morphological Decomposition. IEE Proc. - Vision, Image and Signal Processing **144** (1997) 171-179

[13] Madiraju, S.V.R., Liu, C.C.: Rotation Invariant Texture Classification Using Covariance. Proc. Int. Conf. Image Processing 1 (1995) 262-265, Washington, D.C.

[14] Mao, J., Jain, A.K.: Texture Classification and Segmentation Using Multiresolution Simultaneous Autoregressive Models. Pattern Recognition 25 (1992) 173-188

[15] V. Manian and R. Vasquez, ªScaled and Rotated Texture Classification Using a Class of Basis Functions, Pattern Recognition, vol. 31, pp. 1937-1948, 1998.

[16] Y. Wu and Y. Yoshida, ªAn Efficient Method for Rotation and Scaling Invariant Texture Classification,º Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing, vol. 4, pp. 2519-2522, 1995.

[17] L. Wang and G. Healey, ªUsing Zernike Moments for the Illumination and Geometry Invariant Classification of Multi-Spectral Texture,º IEEE Trans. Image Processing, vol. 7, pp. 196-203,1998

[18] Chen, J.-L., Kundu, A.: Rotation and Gray Scale Transform Invariant Texture Identification Using Wavelet Decomposition and Hidden Markov Model. IEEE Trans. Pattern Analysis and Machine Intelligence **16** (1994) 208-214

[19] Wu, W.-R., Wei, S.-C.: Rotation and Gray-Scale Transform-Invariant Texture Classification Using Spiral Resampling, Subband Decomposition and Hidden Markov Model. IEEE Trans. Image Processing **5** (1996) 1423-1434

[20] Haralick R.M. "Statistical and structural approaches to Texture", *Proc IEEE 67,no.5, 786-804,1979*.

[21] Conners, R.W. and Harlow C.A., "Toward a structural textural analyzer based on statistical methods", *Comput.* Graph, Image Processing,12,224-256, 1980.

[22] Ojala, T., Pietikainen, M. and Maenpaa, T, "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns", *IEEE Trans. Pattern Anal. Mach. Intell, 24, no. 7, 971-987,2002.*

[23] Pun C,-M and Lee, M,-C "Log polar wavelet energy signatures for rotation and scale invariant texture classification", *IEEE Trans. Pattern. Anal, Mach. Intell. 25, no.5, 590-603,2003,*

[24] Ojala T & Pietikäinen M , "Nonparametric multichannel texture description with simple spatial operators", *Proc. 14th International Conference on Pattern Recognition, Brisbane, Australia, 1052 – 1056, 1998.*

[25] Ojala T, Pietikäinen M & Mäenpää T," A generalized Local Binary Pattern operator for multiresolution gray scale and rotation invariant texture classification" , *Advances in Pattern Recognition, ICAPR 2001 Proceedings, Lecture Notes in Computer Science 2013, Springer, 397 - 406. 2001*

[26] Ojala T, Pietikäinen M & Mäenpää T Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7):971-987 2002.

[27] Ahonen T & Pietikäinen M, "A framework for analyzing texture descriptors", Proc. Third International Conference on Computer Vision Theory and Applications (VISAPP 2008), Madeira, Portugal, 1:507-512 2008.

[28] Ojansivu V & Heikkilä J, "A method for blur and affine invariant object recognition using phase-only bispectrum", *Proc. Image Analysis and Recognition (ICIAR 2008), Póvoa de Varzim, Portugal, 5112:527-536. 2008*

[29] Takala V, Ahonen T & Pietikäinen M, "Block-based methods for image retrieval using local binary patterns", *Image Analysis, SCIA 2005 Proceedings, Lecture Notes in Computer Science 3540, Springer, 882-891, 2005.*

Suguna R received M.Tech degree in CSE from IIT Madras, Chennai in 2004. She is currently pursuing the Ph.D. degree in Dept. of IT, MIT Campus, Anna University.

Anandhakumar P received Ph.D degree in CSE from Anna University, in 2006. He is working as Assistant Professor in Dept. of IT, MIT Campus, Anna University. His research area includes image processing and networks Pl. attach color photo