Local Tetra Patterns: A New Feature Descriptor for Content-Based Image Retrieval

Article *in* IEEE Transactions on Image Processing · May 2012

DOI: 10.1109/TIP.2012.2188809 · Source: PubMed

Subrahmanyam Murala
Indian Institute of Technology Ropar
27 PUBLICATIONS **648** CITATIONS
SEE PROFILE

R.P. Maheshwari
Indian Institute of Technology Roorkee
103 PUBLICATIONS **1,459** CITATIONS
SEE PROFILE

Balasubramanian Raman
Indian Institute of Technology Roorkee
190 PUBLICATIONS **1,705** CITATIONS
SEE PROFILE

Local Tetra Patterns: A New Feature Descriptor for Content-Based Image Retrieval

Subrahmanyam Murala, R. P. Maheshwari, *Member, IEEE*, and R. Balasubramanian, *Member, IEEE*

***Abstract*—In this paper, we propose a novel image indexing and retrieval algorithm using local tetra patterns (LTrPs) for content-based image retrieval (CBIR). The standard local binary pattern (LBP) and local ternary pattern (LTP) encode the relationship between the referenced pixel and its surrounding neighbors by computing gray-level difference. The proposed method encodes the relationship between the referenced pixel and its neighbors, based on the directions that are calculated using the first-order derivatives in vertical and horizontal directions. In addition, we propose a generic strategy to compute nth-order LTrP using $(n-1)$th-order horizontal and vertical derivatives for efficient CBIR and analyze the effectiveness of our proposed algorithm by combining it with the Gabor transform. The performance of the proposed method is compared with the LBP, the local derivative patterns, and the LTP based on the results obtained using benchmark image databases viz., Corel 1000 database (DB1), Brodatz texture database (DB2), and MIT VisTex database (DB3). Performance analysis shows that the proposed method improves the retrieval result from 70.34%/44.9% to 75.9%/48.7% in terms of average precision/average recall on database DB1, and from 79.97% to 85.30% and 82.23% to 90.02% in terms of average retrieval rate on databases DB2 and DB3, respectively, as compared with the standard LBP.**

***Index Terms*—Content-based image retrieval (CBIR), Gabor transform (GT), local binary pattern (LBP), local tetra patterns (LTrPs), texture.**

I. Introduction

A. Motivation

THE EXPLOSIVE growth of digital libraries due to Web cameras, digital cameras, and mobile phones equipped with such devices is making the database management by human annotation an extremely tedious and clumsy task. Thus, there exists a dire need for developing an efficient expert technique that can automatically search the desired image from the huge database. Content-based image retrieval (CBIR) is one of the commonly adopted solutions for such applications. The feature extraction in CBIR is a prominent step whose effectiveness depends upon the method adopted for extracting features from given images. The CBIR utilizes visual contents of an image such as color, texture, shape, faces, spatial layout, etc., to represent and index the image database. These features can be further classified as general features such as color, texture, and shape, and domain-specific features such as human faces, fingerprints, etc. The difficulty to find a single best representation of an image for all perceptual subjectivity is due to the fact that the user may take photographs in different conditions such as view angle, illumination changes, etc. A comprehensive and extensive literature survey on CBIR is presented in [1]–[4].

Texture analysis has been extensively used in computer vision and pattern recognition applications due to its potential in extracting the prominent features. Moghaddam *et al.* have introduced the concept of wavelet correlogram [5], [6] and have further shown that the performance improvement can be obtained by optimizing the quantization thresholds [7] using genetic algorithm for CBIR application. Texture retrieval is a branch of texture analysis that has attracted wide attention from industries since this is well suited for the identification of products such as ceramic tiles, marble, parquet slabs, etc. Ahmadian *et al.* have used the discrete wavelet transform (DWT) for texture classification [8]. The application of the DWT using generalized Gaussian density with Kullback–Leibler distance has shown to provide efficient results for texture image retrieval [9] and image segmentation [10]. However, the DWT can extract only three directional (horizontal, vertical, and diagonal) information from an image. To address this directional limitation, Gabor transform (GT) [11], rotated wavelet filters [12], dual-tree complex wavelet filters (DT-CWFs), DT rotated CWFs [13], and rotational invariant complex wavelet filters [14] have been proposed for texture image retrieval.

B. Related Work

The local binary pattern (LBP) feature has emerged as a silver lining in the field of texture classification and retrieval. Ojala *et al.* proposed LBPs [15], which are converted to a rotational invariant version for texture classification [16], [17]. Various extensions of the LBP, such as LBP variance with global matching [18], dominant LBPs [19], completed LBPs [20], joint distribution of local patterns with Gaussian mixtures [21], etc., are proposed for rotational invariant texture classification.

Manuscript received June 11, 2011; revised September 23, 2011, January 14, 2012, and February 09, 2012; accepted February 14, 2012. Date of publication April 03, 2012; date of current version April 18, 2012. This work was supported by the Ministry of Human Resource and Development India under Grant MHR-02-23-200 (429). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Bulent Sankur.

S. Murala is with the Instrumentation and Signal Processing Laboratory, Department of Electrical Engineering, Indian Institute of Technology Roorkee, Roorkee 247667, India (e-mail: subbumurala@gmail.com; subbudee@iitr.ernet.in).

R. P. Maheshwari is with the Department of Electrical Engineering, Indian Institute of Technology Roorkee, Roorkee 247667, India (e-mail: rudrafee@iitr.ernet.in; rpmaheshwari@gmail.com).

R. Balasubramanian is with the Department of Mathematics, Indian Institute of Technology Roorkee, Roorkee 247667, India (e-mail: balarfma@iitr.ernet.in).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2012.2188809

The LBP operator on facial expression analysis and recognition is successfully reported in [22] and [23]. Xi Li *et al.* proposed a multiscale heat-kernel-based face representation as heat kernels are known to perform well in characterizing the topological structural information of face appearance. Furthermore, the LBP descriptor is incorporated into multiscale heat-kernel face representation for the purpose of capturing texture information of the face appearance [24]. Zhang *et al.* proposed local derivative patterns (LDPs) for face recognition, where they considered the LBP as nondirectional first-order local patterns collected from the first-order derivatives and extended the same approach for nth order LDPs [25]. Lei *et al.* [26] proved that exploiting the image information jointly in image space, scale, and orientation domains can provide richer clues, which are not evident in any one individual domain. This process involves two phases. In the first phase, the face image is decomposed into different scale and orientation responses by convolving with multiscale and multiorientation Gabor filters. In the second phase, LBP analysis is used to describe the neighboring relationship not only in image space but also in different scale and orientation responses.

Zhao *et al.* proposed a local spatiotemporal descriptor using the LBP to represent and recognize spoken isolated phrases solely based on visual input [27]. Spatiotemporal LBPs extracted from mouth regions are used for describing isolated phrase sequences. Su *et al.* proposed the hybrid technique for graphic retrieval with the LBP and the Haar wavelet referred as structured local binary Haar pattern that encodes the polarity rather than the magnitude of the difference between accumulated gray values of adjacent rectangles [28]. The LBP has been also used for texture segmentation [29], background modeling and detection [30], shape localization [31], interest region description [32], and biomedical image retrieval [33]. The versions of the LBP and the LDP in the open literature cannot adequately deal with the range of appearance variations that commonly occur in unconstrained natural images due to illumination, pose, facial expression, aging, partial occlusions, etc. In order to address this problem, the local ternary pattern (LTP) [34] has been introduced for face recognition under different lighting conditions.

The LBP, the LDP, and the LTP extract the information based on the distribution of edges, which are coded using only two directions (positive direction or negative direction). Thus, it is evident that the performance of these methods can be improved by differentiating the edges in more than two directions. This observation has motivated us to propose the four direction code, referred to as local tetra patterns (LTrPs) for CBIR.

C. Main Contribution

In this paper, we propose a second-order LTrP that is calculated based on the direction of pixels using horizontal and vertical derivatives. Our method is different from the existing LDP in a manner that it makes use of 0° and 90° derivatives of LDPs for further calculating the directionality of each pixel. The performance resulting from the combination of the GT and the LTrP has been also analyzed. Finally, the generalized nth-order LTrP operator has been presented by using $(n-1)$th-order derivatives. The performance of our method is compared with the LBP, the LDP, and the LTP by conducting three experiments on different image database. Similar to LDP, in order to compare our method with the LBP, we consider the LBP as a nondirectional first-order local pattern called the *first-order* LTrP.

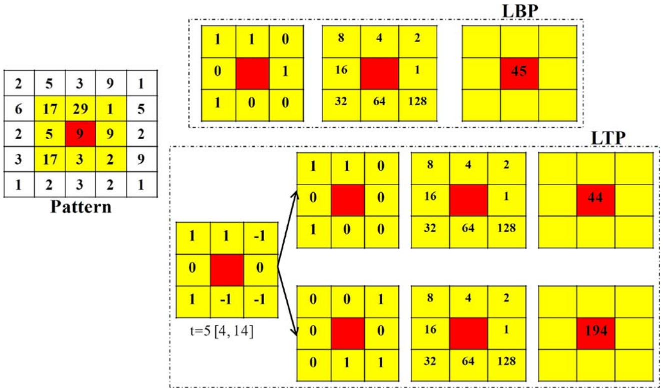


Fig. 1. Calculation of the LBP and LTP operators. In the LTP, the obtained ternary pattern is further coded into upper and lower binary patterns. The upper pattern is obtained by retaining 1 and replacing 0 for -1 and 0. Lower pattern is coded by replacing -1 with 1 and 0 for 1 and 0.

The organization of this paper is as follows: In Section I, a brief review of the texture image retrieval and of related work is given. Section II presents a concise review of local patterns (the LBP, the LDP, the LTP, and the LTrP). Section III presents the concept of multiscale feature extraction, proposed system framework, and query matching. Experimental results and discussions are presented in Section IV, and finally, in Section V, we conclude with the summary of this paper and indicate possible future work.

II. Local Patterns

A. LBPs

The LBP operator was introduced by in [16] for texture classification. Given a center pixel in the image, the LBP value is computed by comparing its gray value with its neighbors, as shown in Fig. 1, based on

$$\mathrm{LBP}_{P,R} = \sum_{p=1}^{P} 2^{(p-1)} \times f_1(g_p - g_c) \tag{1}$$

$$f_1(x) = \begin{cases} 1, & x \geq 0 \\ 0, & \text{else} \end{cases} \tag{2}$$

where g_c is the gray value of the center pixel, g_p is the gray value of its neighbors, P is the number of neighbors, and R is the radius of the neighborhood.

B. LTPs

Tan and Triggs [34] extended the LBP to a three-valued code called the LTP, in which gray values in the zone of width $\pm t$ around g_c are quantized to zero, those above $(g_c + t)$ are quantized to $+1$, and those below $(g_c - t)$ are quantized to -1, i.e., indicator $f_1(x)$ is replaced with three-valued function (3) and the binary LBP code is replaced by a ternary LTP code, as shown in Fig. 1, i.e.,

$$\hat{f}_1(x, g_c, t) = \left\{ \begin{array}{ll} +1, & x \geq g_c + t \\ 0, & |x - g_c| < t \\ -1, & x \leq g_c - t \end{array} \right|_{x=g_p}. \tag{3}$$

More details about LTP can be found in [34].

C. LDPs

Zhang *et al.* proposed the LDPs for face recognition [25]. They considered the LBP as the nondirectional first-order local pattern operator and extended it to higher orders (nth-order) called the LDP. The LDP contains more detailed discriminative features as compared with the LBP.

To calculate the nth-order LDP, the $(n-1)$th-order derivatives are calculated along 0°, 45°, 90°, and 135° directions, denoted as $I_\alpha^{(n-1)}(g_c)|_{\alpha=0^\circ,45^\circ,90^\circ,135^\circ}$. Finally, nth-order LDP is calculated as

$$\mathrm{LDP}_\alpha^n(g_c) = \sum_{p=1}^{P} 2^{(p-1)} \times f_2\left(I_\alpha^{(n-1)}(g_c), I_\alpha^{(n-1)}(g_p)\right)\Bigg|_{P=8} \tag{4}$$

$$f_2(x, y) = \begin{cases} 1, & \text{if } x.y \leq 0 \\ 0, & \text{else.} \end{cases} \tag{5}$$

The detailed discussion and the sample example for the LDP calculation are available in [25].

D. LTrPs

The idea of local patterns (the LBP, the LDP, and the LTP) proposed in [16], [25], and [34] has been adopted to define LTrPs. The LTrP describes the spatial structure of the local texture using the direction of the center gray pixel g_c.

Given image I, the first-order derivatives along 0° and 90° directions are denoted as $I_\theta^1(g_p)|_{\theta=0^\circ,90^\circ}$. Let g_c denote the center pixel in I, and let g_h and g_v denote the horizontal and vertical neighborhoods of g_c, respectively. Then, the first-order derivatives at the center pixel g_c can be written as

$$I_{0^\circ}^1(g_c) = I(g_h) - I(g_c) \tag{6}$$

$$I_{90^\circ}^1(g_c) = I(g_v) - I(g_c) \tag{7}$$

and the direction of the center pixel can be calculated as

$$I_{\mathrm{Dir.}}^1(g_c) = \begin{cases} 1, & I_{0^\circ}^1(g_c) \geq 0 \text{ and } I_{90^\circ}^1(g_c) \geq 0 \\ 2, & I_{0^\circ}^1(g_c) < 0 \text{ and } I_{90^\circ}^1(g_c) \geq 0 \\ 3, & I_{0^\circ}^1(g_c) < 0 \text{ and } I_{90^\circ}^1(g_c) < 0 \\ 4, & I_{0^\circ}^1(g_c) \geq 0 \text{ and } I_{90^\circ}^1(g_c) < 0. \end{cases} \tag{8}$$

From (8), it is evident that the possible direction for each center pixel can be either 1, 2, 3, or 4, and eventually, the image is converted into four values, i.e., directions.

The second-order $\mathrm{LTrP}^2(g_c)$ is defined as

$$\begin{aligned} &\mathrm{LTrP}^2(g_c) \\ &\quad = \left\{ f_3\left(I_{\mathrm{Dir.}}^1(g_c), I_{\mathrm{Dir.}}^1(g_1)\right), f_3\left(I_{\mathrm{Dir.}}^1(g_c), I_{\mathrm{Dir.}}^1(g_2)\right), \right. \\ &\qquad \left. \ldots, f_3\left(I_{\mathrm{Dir.}}^1(g_c), I_{\mathrm{Dir.}}^1(g_P)\right) \right\}\Big|_{P=8} \end{aligned} \tag{9}$$

$$\begin{aligned} &f_3\left(I_{\mathrm{Dir.}}^1(g_c), I_{\mathrm{Dir.}}^1(g_p)\right) \\ &\quad = \begin{cases} 0, & I_{\mathrm{Dir.}}^1(g_c) = I_{\mathrm{Dir.}}^1(g_p) \\ I_{\mathrm{Dir.}}^1(g_p), & \text{else.} \end{cases} \end{aligned} \tag{10}$$

From (9) and (10), we get 8-bit tetra pattern for each center pixel. Then, we separate all patterns into four parts based on the direction of center pixel. Finally, the tetra patterns for each part (direction) are converted to three binary patterns.

Let the direction of center pixel $(I_{\mathrm{Dir.}}^1(g_c))$ obtained using (8) be "1"; then, LTrP^2 can be defined by segregating it into three binary patterns as follows:

$$\begin{aligned} &\mathrm{LTrP}^2|_{\mathrm{Direction}=2,3,4} \\ &\quad = \sum_{p=1}^{P} 2^{(p-1)} \times f_4\left(\mathrm{LTrP}^2(g_c)\right)\Bigg|_{\mathrm{Direction}=2,3,4} \end{aligned} \tag{11}$$

$$\begin{aligned} &f_4\left(\mathrm{LTrP}^2(g_c)\right)|_{\mathrm{Direction}=\phi} \\ &\quad = \begin{cases} 1, & \text{if } \mathrm{LTrP}^2(g_c) = \phi \\ 0, & \text{else} \end{cases} \end{aligned} \tag{12}$$

where $\phi = 2, 3, 4$.

Similarly, the other three tetra patterns for remaining three directions (parts) of center pixels are converted to binary patterns. Thus, we get 12 (4 × 3) binary patterns.

Guo *et al.* [20] used the magnitude component of the local difference operator to propose the magnitude LBP, along with the sign LBP for texture classification. They proved that, although the sign component extracts more useful information as compared with the magnitude component, exploiting the combination of sign and magnitude components can provide better clues, which are not evident in any one individual component. This concept has motivated us to propose the 13th binary pattern (LP) by using the magnitudes of horizontal and vertical first-order derivatives using

$$M_{I^1(g_p)} = \sqrt{\left(I_{0^\circ}^1(g_p)\right)^2 + \left(I_{90^\circ}^1(g_p)\right)^2} \tag{13}$$

$$\mathrm{LP} = \sum_{p=1}^{P} 2^{(p-1)} \times f_1\left(M_{I^1(g_p)} - M_{I^1(g_c)}\right)\Bigg|_{P=8}. \tag{14}$$

For the local pattern with P neighborhoods, 2^P combinations of LBPs are possible, resulting in the feature vector length of 2^P. The computational cost of this feature vector is very high. In order to reduce the computational cost, we consider the uniform patterns [18]. The uniform pattern refers to the uniform appearance pattern that has limited discontinuities in the circular binary representation. In this paper, those patterns that have less than or equal to two discontinuities in the circular binary representation are referred to as the uniform patterns, and the remaining patterns are referred to as nonuniform. Thus, the distinct uniform patterns for a given query image would be $P(P-1)+2$. The possible uniform patterns for $P = 8$ can be seen in [18].

After identifying the local pattern PTN (the LBP, the LTP, the LDP, or the 13-binary-pattern form LTrP), the whole image is represented by building a histogram using

$$\begin{aligned} H_S(l) &= \frac{1}{N_1 \times N_2} \sum_{j=1}^{N_1} \sum_{k=1}^{N_2} f_5\left(\mathrm{PTN}(j,k), l\right); \\ & \qquad l \in [0, P(P-1)+2] \end{aligned} \tag{15}$$

$$f_5(x, y) = \begin{cases} 1, & \text{if } x = y \\ 0, & \text{else} \end{cases} \tag{16}$$

where $N_1 \times N_2$ represents the size of the input image.

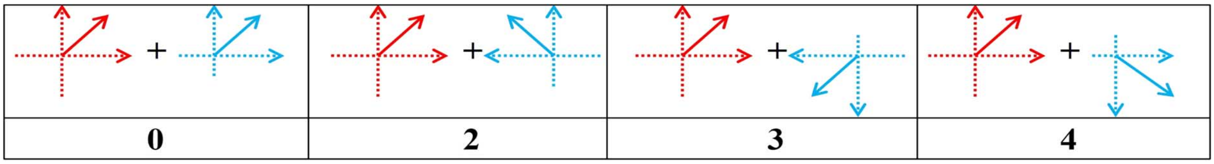
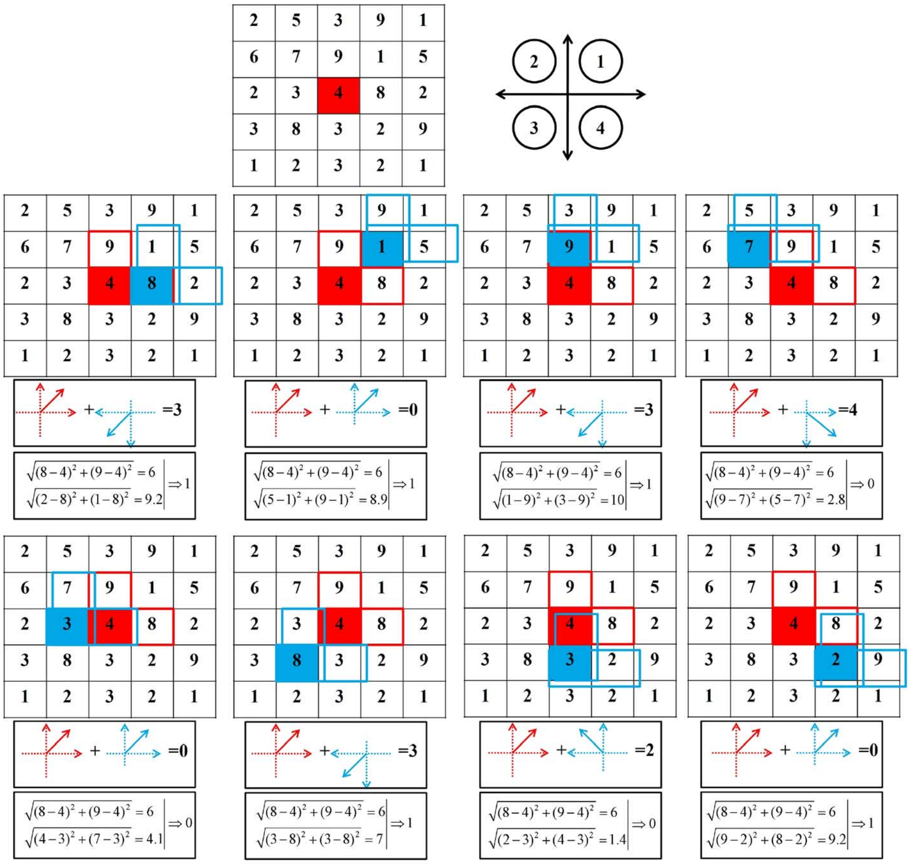


Fig. 2. Calculation of tetra pattern bits for the center-pixel direction "1" using the direction of neighbors. Direction of (red) the center pixel and (cyan) that its neighborhood pixels.


2 5 3 9 1
6 7 9 1 5
2 3 4 8 2
3 8 3 2 9
1 2 3 2 1
1
2
3
4
=3
=0
=3
=4
$\sqrt{(8-4)^2+(9-4)^2}=6$, $\sqrt{(2-8)^2+(1-8)^2}=9.2$ ⇒1
$\sqrt{(8-4)^2+(9-4)^2}=6$, $\sqrt{(5-1)^2+(9-1)^2}=8.9$ ⇒1
$\sqrt{(8-4)^2+(9-4)^2}=6$, $\sqrt{(1-9)^2+(3-9)^2}=10$ ⇒1
$\sqrt{(8-4)^2+(9-4)^2}=6$, $\sqrt{(9-7)^2+(5-7)^2}=2.8$ ⇒0
=0
=3
=2
=0
$\sqrt{(8-4)^2+(9-4)^2}=6$, $\sqrt{(4-3)^2+(7-3)^2}=4.1$ ⇒0
$\sqrt{(8-4)^2+(9-4)^2}=6$, $\sqrt{(3-8)^2+(3-8)^2}=7$ ⇒1
$\sqrt{(8-4)^2+(9-4)^2}=6$, $\sqrt{(2-3)^2+(4-3)^2}=1.4$ ⇒0
$\sqrt{(8-4)^2+(9-4)^2}=6$, $\sqrt{(9-2)^2+(8-2)^2}=9.2$ ⇒1

Tetra Pattern = 3 0 3 4 0 3 2 0

Pattern1 = 0 0 0 0 0 0 1 0; Pattern2 = 1 0 1 0 0 1 0 0; Pattern3 = 0 0 0 1 0 0 0 0

Magnitude Pattern = 1 1 1 0 0 1 0 1

Fig. 3. Example to obtain the tetra and magnitude patterns. For generating a tetra pattern, the bit is coded with the direction of neighbor when the direction of the center pixel and its neighbor are different, otherwise "0." For the magnitude pattern, the bit is coded with "1" when the magnitude of the center pixel is less than the magnitude of its neighbor, otherwise "0."

Fig. 2 illustrates the possible local pattern transitions resulting in an LTrP for direction "1" of the center pixel. The LTrP is coded to "0" when it is equal to the direction of center pixel, otherwise coded in the direction of neighborhood pixel. Using the same analogy, LTrPs are calculated for center pixels having directions 2, 3, and 4.

An example of the second-order LTrP computation resulting in direction "1" for a center pixel marked with red has been illustrated in Fig. 3. When we apply first-order derivative in horizontal and vertical directions to the neighborhood pixel "8," we obtain direction "3" and magnitude "9.2." Since the direction of the center pixel and the direction obtained from the neighborhood pixel are not same, we assign value "3" to the corresponding bit of the LTrP. It can be seen that the magnitude of the center pixel is "6," which is less than the magnitude of neighborhood pixel. Hence, we assign value "1" to the corresponding bit of the magnitude pattern. Similarly, the remaining bits of the LTrP and the magnitude pattern for the other seven neighbors are computed resulting in the tetra pattern "3 0 3 4 0 3 2 0" and the magnitude binary pattern "1 1 1 0 0 1 0 1." After coding the tetra pattern, we separate it into three binary patterns as follows. Referring to the generated LTrP, the first pattern is obtained by keeping "1" where the tetra pattern value is "2" and "0" for other values, i.e., "0 0 0 0 0 0 1 0." Similarly, the other two binary patterns "1 0 1 0 0 1 0 0" and "0 0 0 1 0 0 0 0" are computed for tetra pattern values "3" and "4," respectively.

In the same way, tetra patterns for center pixels having directions 2, 3, and 4 are computed. Thus, with four tetra patterns, 12 binary patterns are obtained. The 13th binary pattern is obtained from the magnitude of the first-order derivatives.

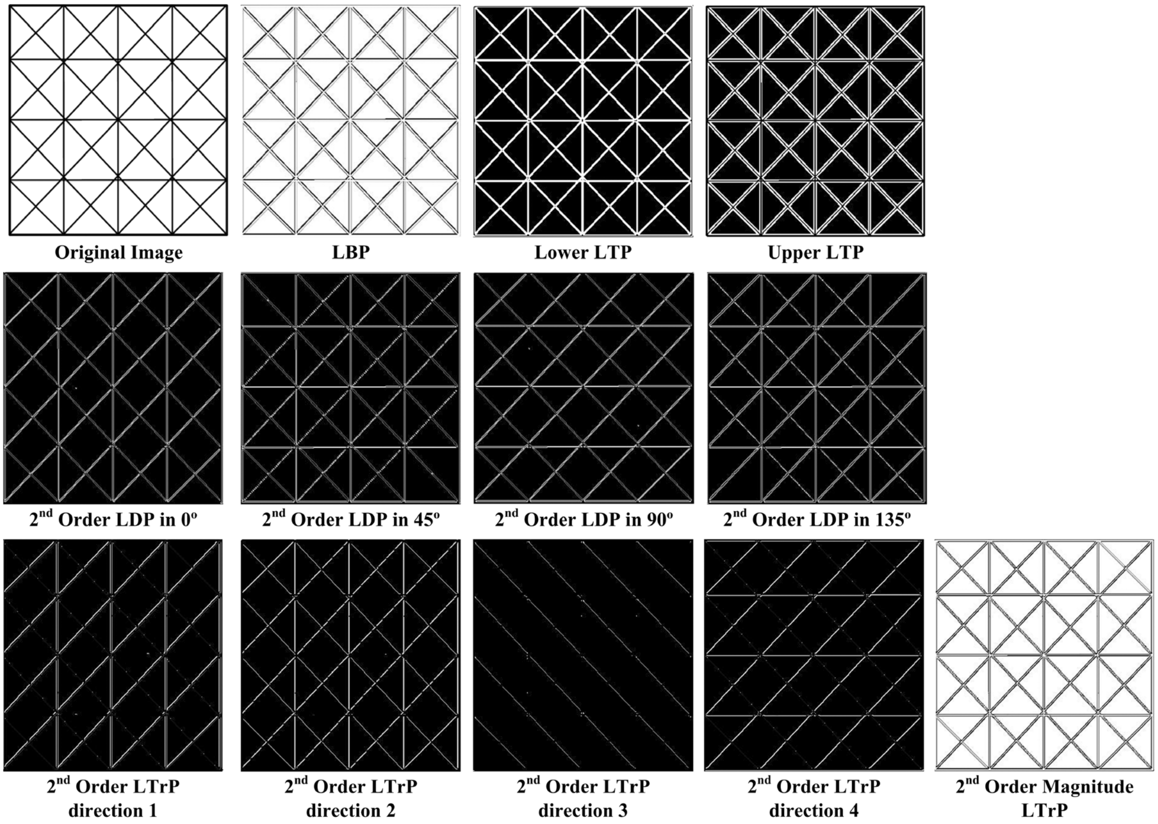


Fig. 4. Each pixel of the original image is considered as the center pixel, and it is coded by using the LBP, LTP, LDP, and LTrP descriptors with the help of neighbors. The possible values of various descriptors are in between 0 to 255, which are further used for dimension reduction by uniform patterns. The LTrP extracts additional directional information, as compared with other patterns.

E. *Nth-Order LTrPs*

To calculate the third-order tetra pattern, initially, the second-order derivatives along horizontal and vertical directions, denoted as $I_{\theta}^{2}(g_p)|_{\theta=0^\circ,90^\circ}$, are computed. Using these derivatives, the third-order tetra pattern $\mathrm{LTrP}^3(g_c)$ is defined as

$$\begin{aligned}\mathrm{LTrP}^3(g_c) = \big\{ & f_3\left(I_{\mathrm{Dir.}}^{2}(g_c), I_{\mathrm{Dir.}}^{2}(g_1)\right), \\ & f_3\left(I_{\mathrm{Dir.}}^{2}(g_c), I_{\mathrm{Dir.}}^{2}(g_2)\right), \\ & \ldots, f_3\left(I_{\mathrm{Dir.}}^{2}(g_c), I_{\mathrm{Dir.}}^{2}(g_P)\right)\big\}\big|_{P=8}. \end{aligned} \tag{17}$$

The generalized formulation for the nth-order LTrP can be defined by using $(n-1)$th-order derivatives in horizontal and vertical directions $I_{\theta}^{n-1}(g_p)|_{\theta=0^\circ,90^\circ}$ as

$$\begin{aligned}\mathrm{LTrP}^n(g_c) = \big\{ & f_3\left(I_{\mathrm{Dir.}}^{n-1}(g_c), I_{\mathrm{Dir.}}^{n-1}(g_1)\right), \\ & f_3\left(I_{\mathrm{Dir.}}^{n-1}(g_c), I_{\mathrm{Dir.}}^{n-1}(g_2)\right), \\ & \ldots, f_3\left(I_{\mathrm{Dir.}}^{n-1}(g_c), I_{\mathrm{Dir.}}^{n-1}(g_P)\right)\big\}\big|_{P=8}. \end{aligned} \tag{18}$$

The higher order LTrPs have the capability to extract more detailed information as compared with lower order LTrPs. However, it has been observed that the second-order LTrP provides better performance as compared with higher order LTrPs (refer to Section IV). This is primarily because higher order LTrPs are more sensitive to noise.

Fig. 4 illustrates the results obtained by applying the LBP, LTP, LDP, and LTrP operators on a reference image. The reference image has been chosen since it provides the results that are visibly comprehensible to differentiate the effectiveness of these approaches. It can be observed that the second-order LTrP operator is able to extract detailed directional information, as compared with second-order LDP, LBP, and LTP operators.

F. *Advantages of the LTrP Over Other Patterns*

The advantages of the LTrP over the LBP, the LDP, and the LTP can be justified with the help of three points.

1) The LBP, the LDP, and the LTP are able to encode images with only two (either “0” or “1”), two (either “0” or “1”), and three (“0,” “1,” or “–1”) distinct values, respectively. However, the LTrP is able to encode images with four distinct values as it is able to extract more detailed information (refer to Fig. 4).
2) The LBP and the LTP encode the relationship between the gray value of the center pixel and its neighbors, whereas the LTrP encodes the relationship between the center pixel and its neighbors based on directions that are calculated with the help of $(n-1)$th-order derivatives.
3) The LDP encodes the relationship between the $(n-1)$th-order derivatives of the center pixel and its neighbors in 0°, 45°, 90°, and 135° directions separately, whereas the LTrP encodes the relationship based on the direction of the center pixel and its neighbors, which are calculated by combining $(n-1)$th-order derivatives of the 0° and 90° directions.

III. Multiscale Feature Extraction

The GT can provide good directional information for texture analysis. Furthermore, the LDP, LTP, and LBP methods also make use of the GT to analyze the effectiveness of their methods for applications in pattern recognition. Thus, in order to compare our results with the aforementioned methods, we have presented the analysis of our method (the LTrP) with the GT.

A. GT

Subrahmanyam *et al.* [35] have given the spatial implementation of the GT. A 2-D Gabor function is a Gaussian modulated by a complex sinusoid. It can be specified by the frequency of sinusoid ω and the standard deviations σ_x and σ_y of the Gaussian envelope as follows:

$$\psi(x,y)=\frac{1}{2\pi\sigma_x\sigma_y}e^{\left[-(1/2)\left(x^2/\sigma_x^2+y^2/\sigma_y^2\right)+2\pi j\omega x\right]}. \tag{19}$$

The response of the Gabor filter is the convolution of the Gabor window with image I and is given by

$$G_{mn}(x,y)=\sum_s\sum_t I\left((x-s,y-t)\psi^*_{mn}(s,t)\right. . \tag{20}$$

B. GLTrPs

In the proposed method, the GT with three scales m and two directions 0° and 90° are applied on a given image to compute derivatives in horizontal and vertical directions.

The nth-order Gabor LTrPs (GLTrPs) is obtained by performing the $(n-1)$th-order derivatives on the real part of the GT response as follows:

$$G^{n-1}_{m,0^\circ}(g_c)=G^{n-2}_{m,0^\circ}(g_c)-G^{n-2}_{m,0^\circ}(g_h),\quad m=1,2,3 \tag{21}$$

$$G^{n-1}_{m,90^\circ}(g_c)=G^{n-2}_{m,90^\circ}(g_c)-G^{n-2}_{m,90^\circ}(g_v),\quad m=1,2,3. \tag{22}$$

The direction of the center pixel $(G^{n-1}_{m,\mathrm{Dir.}}(g_c))$ is calculated by replacing $I^1_{0^\circ}(g_c)\Rightarrow G^{n-1}_{m,0^\circ}(g_c)$ and $I^1_{90^\circ}(g_c)\Rightarrow G^{n-1}_{m,90^\circ}(g_c)$ in (8).

The nth-order GLTrP can be defined as

$$\begin{aligned}\mathrm{GLTrP}^n(g_c)=\Big\{&f_3\left(G^{n-1}_{m,\mathrm{Dir.}}(g_c),G^{n-1}_{m,\mathrm{Dir.}}(g_1)\right),\\ &f_3\left(G^{n-1}_{m,\mathrm{Dir.}}(g_c),G^{n-1}_{m,\mathrm{Dir.}}(g_2)\right),\\ &\ldots,f_3\left(G^{n-1}_{m,\mathrm{Dir.}}(g_c),G^{n-1}_{m,\mathrm{Dir.}}(g_P)\right)\Big\}\Big|_{P=8}.\end{aligned} \tag{23}$$

Furthermore, the 12 binary patterns per scale (m) are calculated similar to (11), and the 13th binary pattern is calculated similar to (13).

C. Proposed System Framework

Fig. 5 illustrates the flowchart of the proposed image retrieval system and algorithm as given below.

Algorithm:

Input: Query image; Output: Retrieval result

1. Load the image, and convert it into grayscale.

2. Apply the first-order derivatives in horizontal and vertical axis.

3. Calculate the direction for every pixel.

4. Divide the patterns into four parts based on the direction of the center pixel.

5. Calculate the tetra patterns, and separate them into three binary patterns.

6. Calculate the histograms of binary patterns.

7. Calculate the magnitudes of center pixels using (13).

8. Construct the binary patterns, and calculate their histogram.

9. Combine the histograms calculated from steps 6 and 8.

10. Construct the feature vector.

11. Compare the query image with the images in the database using (24).

12. Retrieve the images based on the best matches.

This algorithm is also applied on Gabor wavelet subbands (with three scales and two directions) for GLTrP.

D. Query Matching

The feature vector for the query image Q represented as $f_Q=(f_{Q_1},f_{Q_1},\ldots,f_{Q_{Lg}})$ is obtained from feature extraction. Similarly, each image in the database is represented with the feature vector $f_{\mathrm{DB}_j}=(f_{\mathrm{DB}_{j1}},f_{\mathrm{DB}_{j1}},\ldots,f_{\mathrm{DB}_{jLg}})$, $j=1,2,\ldots,|\mathrm{DB}|$. The goal is to select the n best images that resemble the query image. This involves the selection of n top-matched images by measuring the distance between the query image and the images in database $|\mathrm{DB}|$. In order to match the images, we use d_1 similarity distance metric computed using

$$D(Q,\mathrm{DB})=\sum_{i=1}^{Lg}\left|\frac{f_{\mathrm{DB}_{ji}}-f_{Q_i}}{1+f_{\mathrm{DB}_{ji}}+f_{Q_i}}\right| \tag{24}$$

where $f_{\mathrm{DB}_{ji}}$ is the ith feature of the jth image in database $|DB|$.

IV. Experimental Results and Discussions

In order to analyze the performance of the proposed method, experiments were conducted on three benchmark databases. The results are discussed in the following subsections.

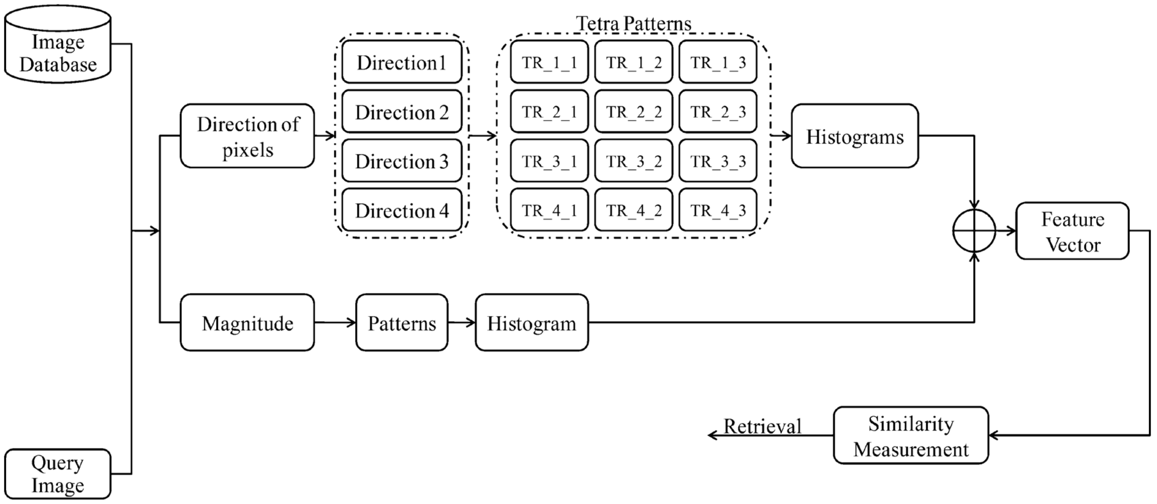


Fig. 5. Proposed image retrieval system framework.

Given below are the abbreviations used in the analysis of the result.

LBP	*LBP features.*
GLBP	*LBP with* GT.
LDP	*Local derivative patterns.*
GLDP	*LDP with GT.*
LTP	*Local ternary patterns.*
GLTP	*LTP with* GT.
LTrP	*Local tetra patterns.*
GLTrP	*LTrP with GT.*
PM1	*LTrP_2nd_order/GLTrP_2nd_order.*
PM2	*LTrP_3rd_order/GLTrP_3rd_order.*
PM3	*LTrP_4th_order/GLTrP_4th_order.*
LDP1	*LDP_2nd_order.*
LDP2	*LDP_3rd_order.*
LDP3	*LTrP_4th_order.*
GLDP1	G*LDP_2nd_order.*
GLDP2	G*LDP_3rd_order.*
GLDP3	G*LTrP_4th_order.*

A. Experiment 1

In experiment 1, images from the Corel database [36] have been used. This database consists of a large number of images of various contents ranging from animals to outdoor sports to natural images. These images have been preclassified into different categories each of size 100 by domain professionals. Some researchers are of the opinion that the Corel database meets all the requirements to evaluate an image retrieval system, due to its large size and heterogeneous content. For our experiment, we have collected 1000 images to form database DB1. These images are collected from ten different domains, namely, *Africans*, *beaches*, *buildings*, *buses*, *dinosaurs*, *elephants*, *flowers*, *horses*, *mountains,* and *food*. Each category has N_G $(= 100)$ images with resolution of either 256 × 384 or 384 × 256. Fig. 6 shows the sample images of DB1 database (one image from each category). In this experiment, each image in the database is used as the query image. For each query, the system collects n database images $X = (x_1, x_2, \ldots, x_n)$ with the shortest image matching distance computed using (24). If the retrieved image $x_i = 1, 2, \ldots., n$ belongs to same category as that of the query image, then we say that the system has appropriately identified the expected image, or else, the system has failed to find the expected image.

Fig. 6. Sample images from database DB1 (one image per category).

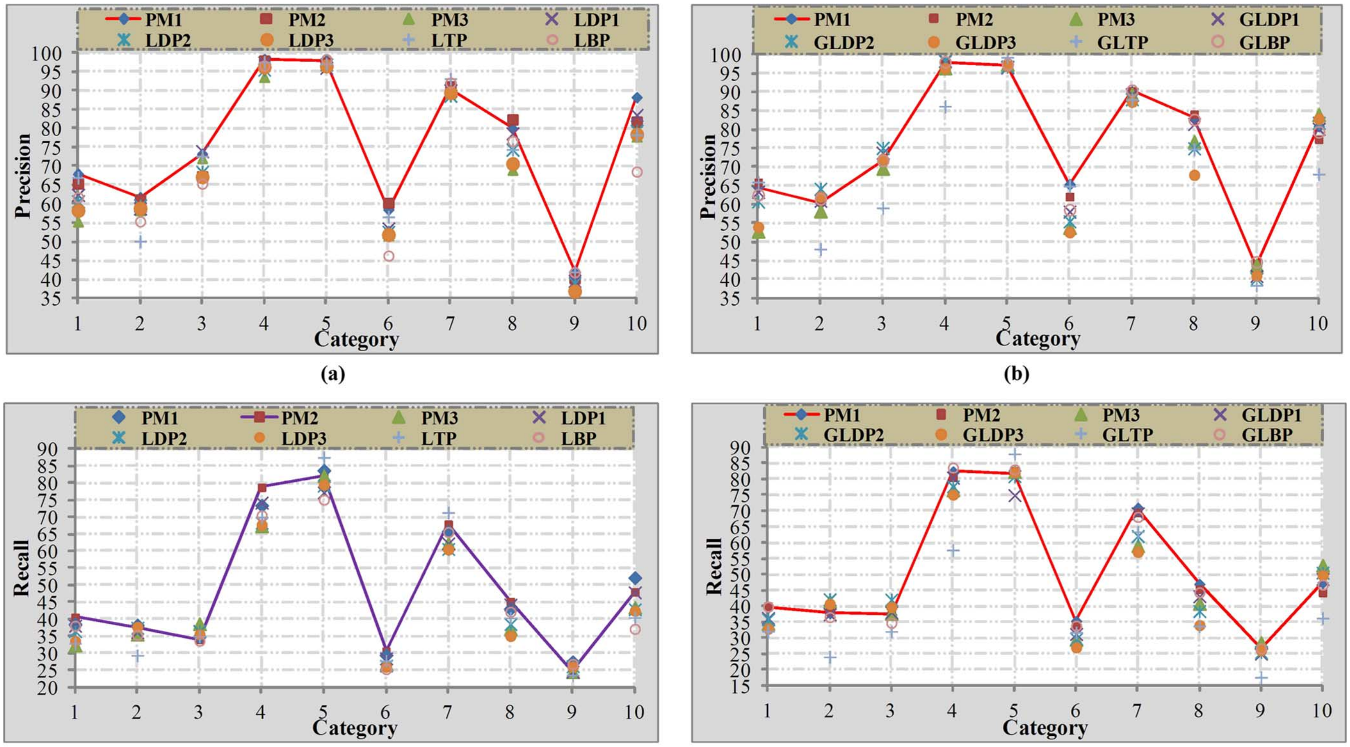
The performance of the proposed method is measured in terms of average precision, average recall, and average retrieval rate (ARR), as shown below.

For the query image I_q, the precision is defined as follows:

$$P(I_q, n) = \frac{1}{n}\sum_{i=1}^{|\mathrm{DB}|} \left|\delta\left(\Phi(I_i), \Phi(I_q)\right)\right| \mathrm{Rank}(I_i, I_q) \leq n| \quad (25)$$

where "n" indicates the number of retrieved images and $|DB|$ is the size of the image database. $\Phi(x)$ is the category of "x," $Rank(I_i, I_q)$ returns the rank of image I_i (for the query image I_q) among all images of $|DB|$, and

$$\delta\left(\Phi(I_i), \Phi(I_q)\right) = \begin{cases} 1, & \Phi(I_i) = \Phi(I_q) \\ 0, & \text{else.} \end{cases}$$



(a) (b) (c) (d)

Fig. 7. Category wise performance comparison of the LTrP/GLTrP with other existing methods on database DB1. The performance in terms of precision (a) without and (b) with GTs. The performance in terms of recall (c) without and (d) with GTs. For the LTrP without GT, PM1: LTrP_2nd_Order; PM2: LTrP_3rd_Order; and PM3: LTrP_4th_Order. For that with GT,: PM1: GLTrP_2nd_Order; PM2: GLTrP_3rd_Order; and PM3: GLTrP_4th_Order.

Recall is defined as

$$R(I_q, n) = \frac{1}{N_G} \sum_{i=1}^{|\mathrm{DB}|} \left|\delta\left(\Phi(I_i), \Phi(I_q)\right) \middle| \mathrm{Rank}(I_i, I_q) \le n\right| . \tag{26}$$

The average precision for the jth similarity category of the reference image database are given by

$$P_{\mathrm{ave}}^{j}(n) = \frac{1}{N_G} \sum_{i \in N_G} P(I_i, n). \tag{27}$$

Finally, the total average precision and the ARR for the whole reference image database are computed using (28) and (29), respectively, i.e.,

$$P_{\mathrm{ave}}^{\mathrm{Total}}(n) = \frac{1}{|\mathrm{DB}|} \sum_{i=1}^{|\mathrm{DB}|} P(I_i, n) \tag{28}$$

$$\mathrm{ARR} = \frac{1}{|\mathrm{DB}|} \sum_{i=1}^{|\mathrm{DB}|} R(I_i, n) \Bigg|_{n \le 100} . \tag{29}$$

The average recall R is also defined in the same manner.

Fig. 7 illustrates category wise results of the LTrP and other existing methods with and without GT in terms of precision and recall on the DB1 database.

The results are considered to be better if average values of precision and recall are high. From Fig. 7, it can be observed that the LTrP/GLTrP outperforms the other methods in more than seven out of ten categories. Fig. 8 shows the experimental results of the LTrP/GLTrP and other existing methods in terms of average precision and ARR versus the number of top matches. Fig. 9 illustrates the comparison between different orders of LTrPs and LDPs with and without GT in terms of average precision and recall. Based on these figures, it is evident that the LTrP/GLTrP significantly improves retrieval results in terms of average precision, average recall, and ARR on image database DB1, as compared with other existing methods.

B. Experiment 2

In experiment 2, database DB2 is used, which consists of 116 different textures. We have used 109 textures from Brodatz texture photographic album [37] and seven textures from the University of Southern California database [38]. The size of each texture is 512 × 512. Each 512 × 512 image is divided into sixteen 128 × 128 nonoverlapping subimages, thus creating a database of 1856 (116 × 16) images. In this experiment, each image in the database is considered as the query image, and the performance of the proposed method is measured in terms of ARR as given by

$$\mathrm{ARR} = \frac{1}{|\mathrm{DB}|} \sum_{i=1}^{|\mathrm{DB}|} R(I_i, n) \Bigg|_{\substack{N_G = 16 \\ n \ge 16}} . \tag{30}$$

Database DB2 is used to compare the performance of the LTrP with other existing methods (the LTP, the LDP, and the LBP) with and without GT. Table I shows the results of all techniques with and without GT in terms of ARR.

Fig. 10 illustrates the retrieval performance of the LTrP/GLTrP and other existing methods as a function of the number of top matches. Fig. 11 shows the comparison

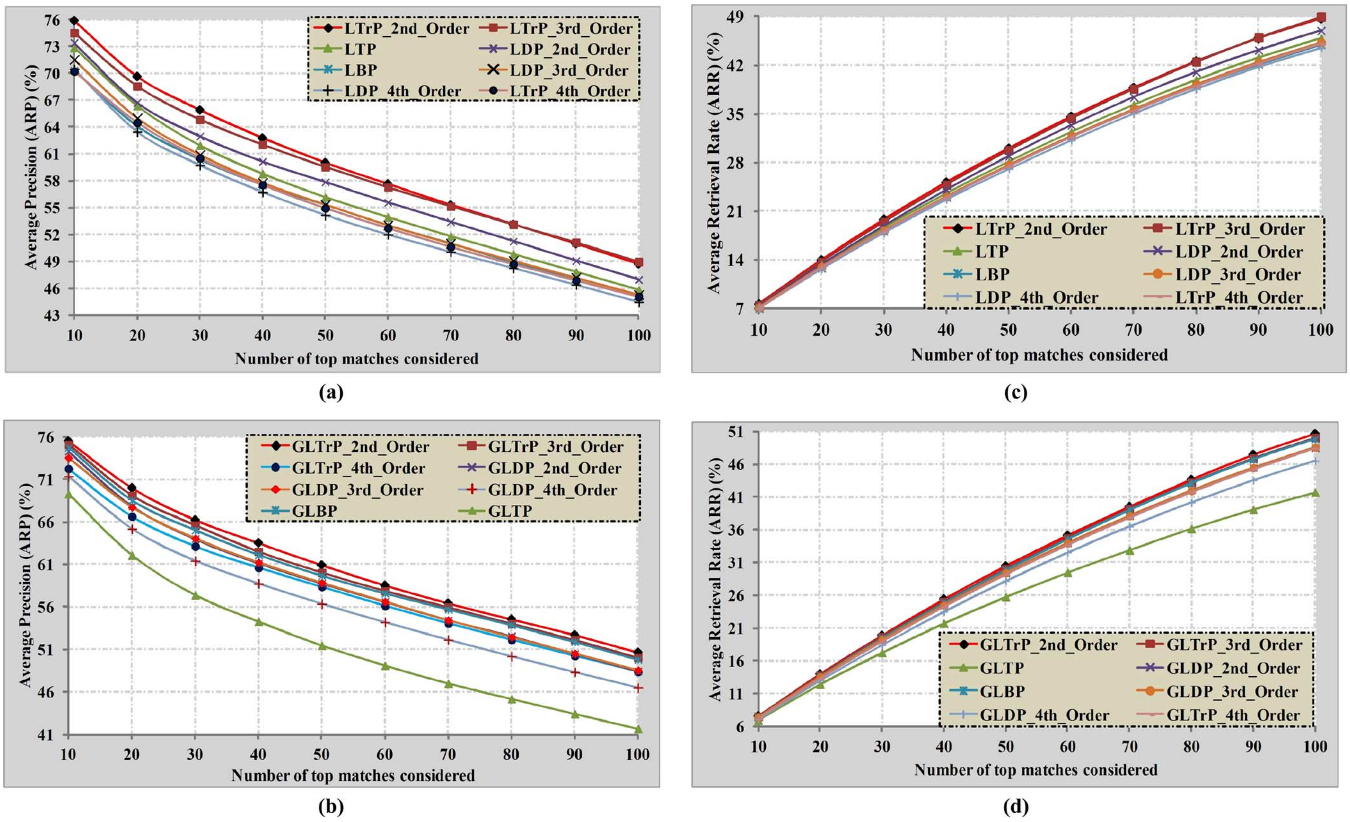


Fig. 8. Comparison of the LTrP/GLTrP with other existing methods in terms of (a) and (b) average retrieval precision and (c) and (d) ARR on database DB1.

TABLE I
ARR FOR THE 116 TEXTURE CLASSES OF DATABASE DB2 AND 40 TEXTURE CLASSES OF DATABASE DB3

Method	ARR (%)	
	DB2	DB3
LTrP_2nd_Order	**85.30**	90.02
LTrP_3rd_Order	83.31	89.70
LTrP_4th_Order	76.78	80.01
LDP_2nd_Order	79.91	87.27
LDP_3rd_Order	79.47	85.48
LDP_4th_Order	77.34	81.58
LTP	82.51	87.52
LBP	79.97	82.23
GLTrP_2nd_Order	81.97	**90.16**
GLTrP_3rd_Order	82.04	89.41
GLTrP_4th_Order	81.52	87.32
GLDP_2nd_Order	79.11	87.51
GLDP_3rd_Order	79.24	88.18
GLDP_4th_Order	78.87	87.61
GLTP	78.75	80.98
GLBP	79.62	87.07

between different orders of LTrPs and LDPs with and without GT in terms of ARR. Based on Table I and Figs. 10 and 11, it is evident that the LTrP/GLTrP outperforms the other existing methods.

C. Experiment 3

In this experiment, database DB3 is used, which consists of 40 different textures collected from the MIT VisTex database [39]. The size of each texture is 512 × 512. Each 512 × 512 image is divided into sixteen 128 × 128 nonoverlapping subimages, thus creating a database of 640 (40 × 16) images. The performance of the proposed method is measured in terms of ARR computed using (30).

TABLE II
FEATURE VECTOR LENGTH OF QUERY IMAGE USING VARIOUS METHODS

Method	Feature Vector Length
LBP	59
LTP	2×59
LDP	4×59
LTrP	13×59

Database DB3 has been used to compare the performance of the LTrP with other existing methods (the LTP, the LDP, and the LBP) with and without GT. From Table I, it is evident that the LTrP/GLTrP outperforms other existing methods. Fig. 12 shows the graphs illustrating the retrieval performance of the LTrP/GLTrP and other existing methods as a function of the number of top matches. Fig. 13 shows the comparison between different orders of LTrPs and LDPs with and without GT in terms of ARR. From Table I and Figs. 12 and 13, it is evident that the LTrP/GLTrP yields better performance as compared with the other existing methods.

D. Feature Vector Length V/s Performance

Table II shows the feature vector length for a given query image using the LBP, the LTP, the LDP, and the LTrP. From Tables I and II and Figs. 7–13, it can be observed that the feature

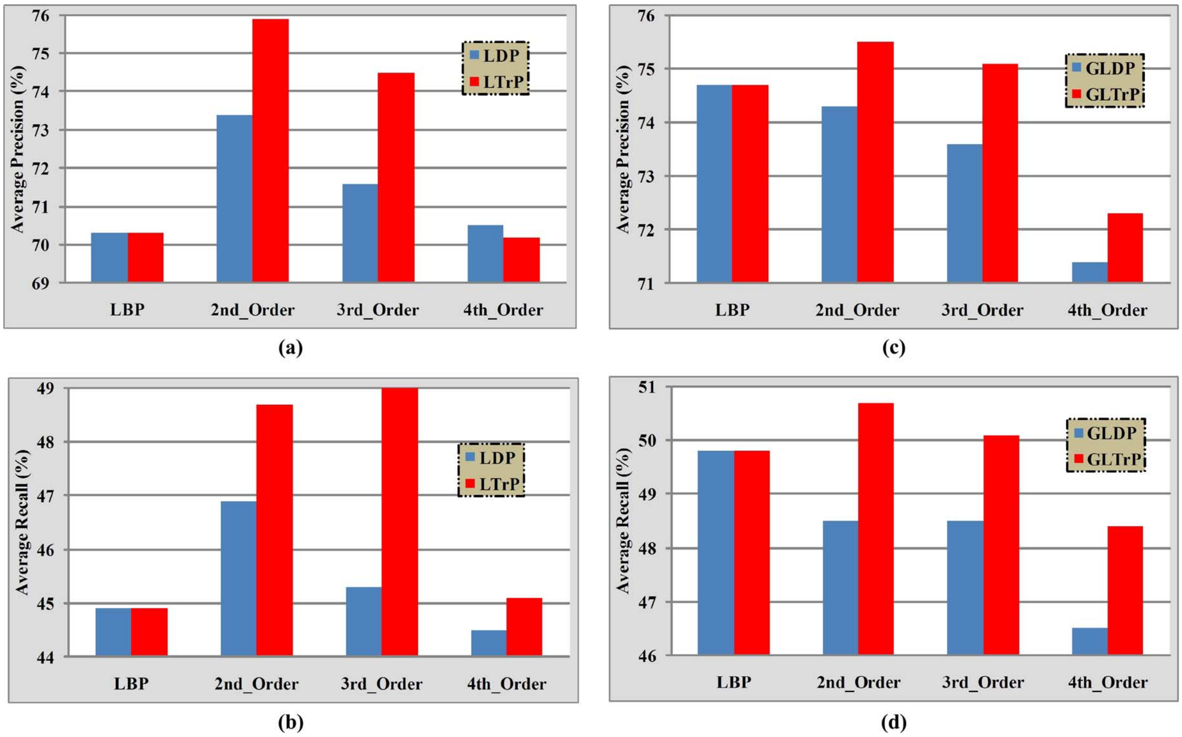


Fig. 9. Comparison between the different orders of the LTrP/GLTrP and LDP/GLDP in terms of (a) and (c) average retrieval precision and (b) and (d) ARR on database DB1.

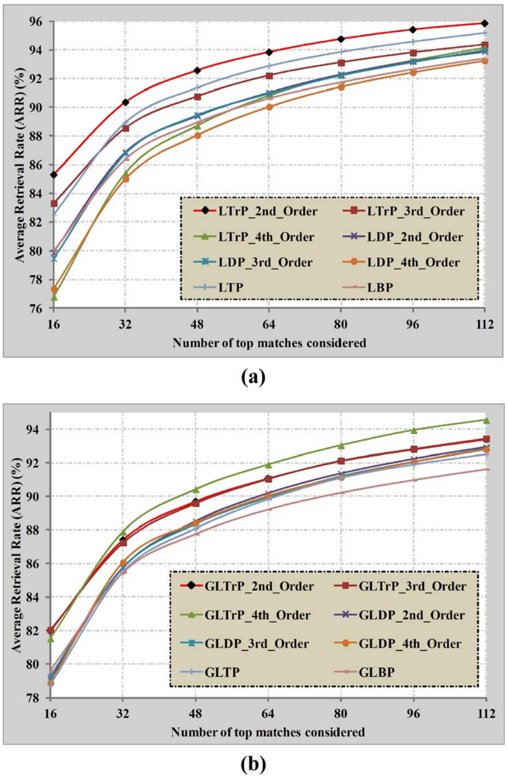


Fig. 10. Comparison of the LTrP with other existing methods in terms of ARR on database DB2. (a) Without and (b) with GT.

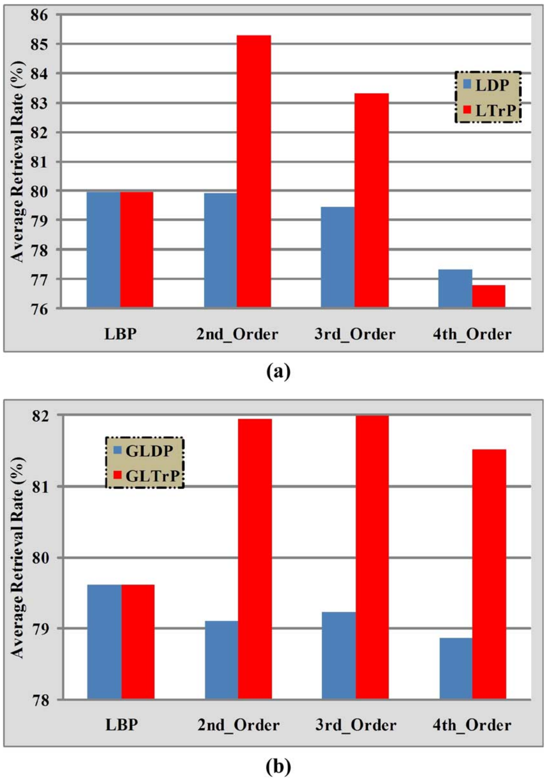
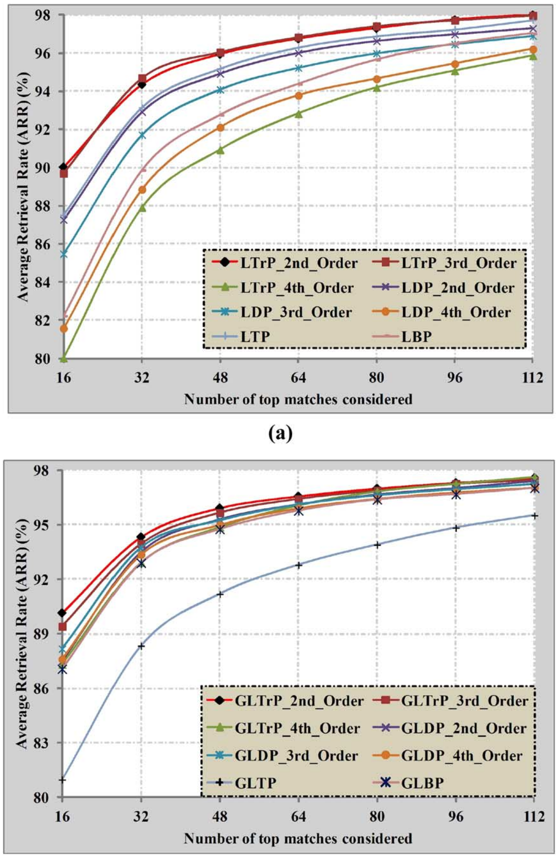


Fig. 11. Comparison between the different orders of the LTrP/GLTrP and the LDP in terms of ARR on database DB2. (a) Without and (b) with GT.

vector length of the LTrP is 13, 6.5, and 3.25 times of the LBP, the LTP, and the LDP, respectively, as it outperforms:

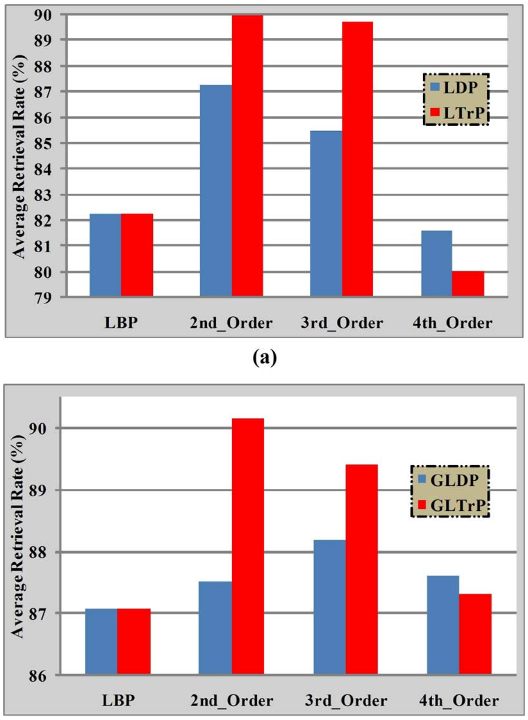
1) the LBP by 5.6%, the LTP by 3%, and the LDP by 2.5% in terms of average precision on database DB1;



(a)

(b)

Fig. 12. Comparison of the LTrP with other existing methods in terms of ARR on database DB3. (a) Without and (b) with GT.



(a)

(b)

Fig. 13. Comparison between the different orders of the LTrP and the LDP in terms of ARR on database DB3. (a) Without and (b) with GT.

2) the LBP by 4.1%, the LTP by 3.2%, and the LDP by 2.1% in terms of average recall on database DB1;
3) the LBP by 5.3%, the LTP by 2.8%, and the LDP by 5.4% in terms of ARR on database DB2;
4) the LBP by 7.8%, the LTP by 2.5%, and the LDP by 2.8% in terms of ARR on database DB3.

V. Conclusion and Future Work

In this paper, we have presented a novel approach referred as LTrPs for CBIR. The LTrP encodes the images based on the direction of pixels that are calculated by horizontal and vertical derivatives. The magnitude of the binary pattern is collected using magnitudes of derivatives. The effectiveness of the proposed approach has been also analyzed by combining it with the GT.

The performance improvement of the proposed method has been compared with the LBP, the LTP, and the LDP on grayscale images and has been detailed below.

1) The average precision has significantly improved from 70.34%, 72.9%, and 73.4% to 75.9%, as compared with the LBP, the LTP, and the LDP, respectively, on database DB1.
2) The average recall has improved from 44.9%, 45.8%, and 46.9% to 48.7%, as compared with the LBP, the LTP, and the LDP, respectively, on database DB1.
3) The ARR has improved from 79.97%, 82.5%, and 79.91% to 85.3%, as compared with the LBP, the LTP, and the LDP, respectively, on database DB2.
4) The ARR has improved from 82.23%, 87.52%, and 87.27% to 90.02%, as compared with the LBP, the LTP, and the LDP, respectively, on database DB3.

In this paper, only horizontal and vertical pixels have been used for derivative calculation. Results can be further improved by considering the diagonal pixels for derivative calculations in addition to horizontal and vertical directions. Due to the effectiveness of the proposed method, it can be also suitable for other pattern recognition applications such as face recognition, fingerprint recognition, etc.

Acknowledgment

The authors would like to thank Mr. A. B. Gonde, Dr. D. K. Rajoriya, and Mr. P. Arvind (Research Scholars), Department of Electrical Engineering, and Mr. D. Gangodkar (Research Scholar), Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, Roorkee, for their valuable technical discussions for this paper. They would also like to thank the associate editor and anonymous reviewers for insightful comments and helpful suggestions to improve the quality, which have been incorporated in this paper.

References

[1] Y. Rui and T. S. Huang, "Image retrieval: Current techniques, promising directions and open issues," *J. Visual Commun. Image Represent.*, vol. 10, no. 1, pp. 39–62, Mar. 1999.

[2] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 12, pp. 1349–1380, Dec. 2000.

[3] M. Kokare, B. N. Chatterji, and P. K. Biswas, "A survey on current content based image retrieval methods," *IETE J. Res.*, vol. 48, no. 3&4, pp. 261–271, 2002.

[4] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma, "A survey of content-based image retrieval with high-level semantics," *Pattern Recogn.*, vol. 40, no. 1, pp. 262–282, Jan. 2007.

[5] H. A. Moghaddam, T. T. Khajoie, and A. H. Rouhi, "A new algorithm for image indexing and retrieval using wavelet correlogram," in *Proc. ICIP*, 2003, pp. III-497–III-500.

[6] H. A. Moghaddam and M. Saadatmand Tarzjan, "Gabor wavelet correlogram algorithm for image indexing and retrieval," in *Proc. ICPR*, 2006, pp. 925–928.

[7] M. Saadatmand Tarzjan and H. A. Moghaddam, "A novel evolutionary approach for optimizing content based image retrieval," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 1, pp. 139–153, Feb. 2007.

[8] A. Ahmadian and A. Mostafa, "An efficient texture classification algorithm using gabor wavelet," in *Proc. EMBS*, 2003, pp. 930–933.

[9] M. N. Do and M. Vetterli, "Wavelet-based texture retrieval using generalized Gaussian density and Kullback–Leibler distance," *IEEE Trans. Image Process.*, vol. 11, no. 2, pp. 146–158, Feb. 2002.

[10] M. Unser, "Texture classification by wavelet packet signatures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 11, pp. 1186–1191, Nov. 1993.

[11] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 8, pp. 837–842, Aug. 1996.

[12] M. Kokare, P. K. Biswas, and B. N. Chatterji, "Texture image retrieval using rotated wavelet filters," *Pattern Recogn. Lett.*, vol. 28, no. 10, pp. 1240–1249, Jul. 2007.

[13] M. Kokare, P. K. Biswas, and B. N. Chatterji, "Texture image retrieval using new rotated complex wavelet filters," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 6, pp. 1168–1178, Dec. 2005.

[14] M. Kokare, P. K. Biswas, and B. N. Chatterji, "Rotation-invariant texture image retrieval using rotated complex wavelet filters," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 6, pp. 1273–1282, Dec. 2006.

[15] T. Ojala, M. Pietikainen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recogn.*, vol. 29, no. 1, pp. 51–59, Jan. 1996.

[16] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[17] M. Pietikainen, T. Ojala, T. Scruggs, K. W. Bowyer, C. Jin, K. Hoffman, J. Marques, M. Jacsik, and W. Worek, "Rotational invariant texture classification using feature distributions," *Pattern Recogn.*, vol. 33, no. 1, pp. 43–52, Jan. 2000.

[18] Z. Guo, L. Zhang, and D. Zhang, "Rotation invariant texture classification using LBP variance with global matching," *Pattern Recogn.*, vol. 43, no. 3, pp. 706–719, Mar. 2010.

[19] S. Liao, M. W. K. Law, and A. C. S. Chung, "Dominant local binary patterns for texture classification," *IEEE Trans. Image Process.*, vol. 18, no. 5, pp. 1107–1118, May 2009.

[20] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1657–1663, Jun. 2010.

[21] H. Lategahn, S. Gross, T. Stehle, and T. Aach, "Texture classification by modeling joint distributions of local patterns with Gaussian mixtures," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1548–1557, Jun. 2010.

[22] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Applications to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 2037–2041, Dec. 2006.

[23] G. Zhao and M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 915–928, Jun. 2007.

[24] X. Li, W. Hu, Z. Zhang, and H. Wang, "Heat Kernel based local binary pattern for face representation," *IEEE Signal Process. Lett.*, vol. 17, no. 3, pp. 308–311, Mar. 2010.

[25] B. Zhang, Y. Gao, S. Zhao, and J. Liu, "Local derivative pattern versus local binary pattern: Face recognition with higher-order local pattern descriptor," *IEEE Trans. Image Process.*, vol. 19, no. 2, pp. 533–544, Feb. 2010.

[26] Z. Lei, S. Liao, M. Pietikäinen, and S. Z. Li, "Face recognition by exploring information jointly in space, scale and orientation," *IEEE Trans. Image Process.*, vol. 20, no. 1, pp. 247–256, Jan. 2011.

[27] G. Zhao, M. Barnard, and M. Pietikäinen, "Lipreading with local spatiotemporal descriptors," *IEEE Trans. Multimedia*, vol. 11, no. 7, pp. 1254–1265, Nov. 2009.

[28] S.-Z. Su, S.-Y. Chen, S.-Z. Li, S.-A. Li, and D.-J. Duh, "Structured local binary Haar pattern for pixel-based graphics retrieval," *Electron. Lett.*, vol. 46, no. 14, pp. 996–998, Jul. 2010.

[29] M. Li and R. C. Staunton, "Optimum Gabor filter design and local binary patterns for texture segmentation," *Pattern Recog.*, vol. 29, no. 5, pp. 664–672, Apr. 2008.

[30] M. Heikkila and M. Pietikainen, "A texture based method for modeling the background and detecting moving objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 657–662, Apr. 2006.

[31] X. Huang, S. Z. Li, and Y. Wang, "Shape localization based on statistical method using extended local binary patterns," in *Proc. ICIG*, 2004, pp. 184–187.

[32] M. Heikkila, M. Pietikainen, and C. Schmid, "Description of interest regions with local binary patterns," *Pattern Recog.*, vol. 42, no. 3, pp. 425–436, Mar. 2009.

[33] D. Unay, A. Ekin, and R. S. Jasinschi, "Local structure-based region-of-interest retrieval in brain MR images," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 4, pp. 897–903, Jul. 2010.

[34] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1635–1650, Jun. 2010.

[35] M. Subrahmanyam, A. B. Gonde, and R. P. Maheshwari, "Color and texture features for image indexing and retrieval," in *Proc. IACC*, 2009, pp. 1411–1416.

[36] Corel 1000 and Corel 10000 image database [Online]. Available: http://wang.ist.psu.edu/docs/related.shtml

[37] P. Brodatz*, Textures: A Photographic Album for Artists and Designers*. New York: Dover, 1996.

[38] University of Southern California, Los Angeles, "Signal and image processing institute," [Online]. Available: http://sipi.usc.edu/database/

[39] MIT Vision and Modeling Group, Cambridge, "Vision texture," [Online]. Available: http://vismod.media.mit.edu/pub/

Subrahmanyam Murala was born in India in 1985. He received the B.E. degree in electrical and electronics engineering from Sagi Ramakrishnam Raju Engineering College, Bhimavaram, India, in 2007, the M.Tech. degree in measurements and instrumentation from the Indian Institute of Technology Roorkee, Roorkee, India, in 2009, where he is currently pursuing the Ph.D. degree in the Department of Electrical Engineering.

His major fields of interests are content-based image retrieval, biomedical imaging, and object tracking.

R. P. Maheshwari (M'07) was born in India in 1960. He received the B.E. degree in electrical engineering and the M.Sc. (engineering) degree in instrumentation from Aligarh Muslim University, Uttar Pradesh, India, in 1982 and 1985, respectively, and the Ph.D. degree from the University of Roorkee [currently the Indian Institute of Technology (IIT) Roorkee], Roorkee, India, in 1996.

He is currently a Professor with the Department of Electrical Engineering, IIT Roorkee. He has published more than 100 research papers in international and national journals and conferences. His fields of interest include wavelets, neural network, optimization techniques, content-based image retrieval, digital instrumentation, digital protective relays, and digital signal processing.

Dr. Maheshwari is a Life Member of the Institution of Engineers (India) and the Institution of Electronics and Telecommunication Engineers (India).

R. Balasubramanian (M'08) received the B.Sc. degree in mathematics from Agurchand Manmull Jain College (University of Madras), Chennai, India, in 1994, the M.Sc. degree in mathematics from Madras Christian College (University of Madras), Madras, India, in 1996, and the Ph.D. degree from the Indian Institute of Technology (IIT) Madras, Chennai, India, in 2001.

During his research tenure with IIT Madras, he was selected with full financial support twice by the International Centre for Theoretical Physics, Trieste, Italy, to attend five weeks of special short-term courses in the field of image processing and control theory. He was a Postdoctoral Fellow with the University of Missouri Columbia from 2001 to 2002 and a Postdoctoral Associate with Rutgers, State University of New Jersey, from 2002 to 2003. In 2004, he was a Lecturer with the Department of Mathematics, IIT Roorkee, Roorkee, India, where he has been an Assistant Professor since 2006. From May to August 2009, he was a Visiting Professor and a Member of the Computer Vision and Sensing Systems Laboratory, Department of Electrical and Computer Engineering, University of Windsor, Canada. His area of research includes vision geometry, content-based image retrieval, digital watermarking using mathematical transformations, image fusion, biometrics, secure image transmission over wireless channel, and hyperspectral imaging.