An Enhanced Technique for Image Indexing and Retrieval with Orientation Features Using Autocorrelation Function

K.Selvarajan*
Ph.D Research Scholar & Associate Professor, Department of Computer Science, Bishop Thorp College, Dharapuram
Email: ks_btc@yahoo.in
Dr.S.Pannirselvam
Research Supervisor & Head
Department of Computer Science, Erode Arts & Science College (Autonomous), Erode, Tamil Nadu, India.
Email: pannirselvam08@gmail.com

Abstract-- In this paper, the rectangular features of an image are considered for generation of feature set and compared with the existing techniques. The orientation features are extracted for the establishment of feature set for image indexing and retrieval. The identification and the representation of textures present in the image under analysis are performed with auto-correlation coefficients by estimating the model parameters for small image regions. The feature is generated and matching is done by measuring distance between two images using Euclidean distance classification. The experiments are carried out on the standard images using MATLAB. The result shows that proposed method provides better retrieval rate when compared with the existing methods such as Local Tetra Patterna and Local Ternary Pattern Method.

Keywords— LBP,CBIR,LTP,ACF,

1. INTRODUCTION

Edge based image retrieval is one of the best approaches for image retrieval system. The extraction of edges in 2D monochrome images based on the proposed Full Range Auto Regressive Model is first presented in this chapter and is extended for subsequent effective image retrieval.

Generally, an edge is defined as a boundary point or outermost area of an object. In image retrieval, edge plays a major role. The retrieval process is done by detecting the edges, computing their values and performing the low-level tasks such as identification, classification, matching, segmentation, compression and boundary detection. In general, edges are characterized by an abrupt change in the gray value of an image and they are classified as roof edges and step edges.

2. LITERATURE SURVEY

In the literature, many reviews on edge detection are reported and various algorithms are proposed for edge extraction in gray level images. Edge detection methods are classified into enhancement /threshold type, gradient based operators, edge-fitting, edge detection, zero-crossings in second order derivatives, curve fitting approach, optimally criterion approach and residual based techniques.

The enhancement/threshold type is well-known for its simplicity and low computational complexity. The gradient values of different orders are calculated by convolving the various gradient operators. The gradient based operators such as Roberts [16], Prewitt [13] and Sobel [21] are well known and widely used to find the step edges rather than the roof edges. All these operators are first order derivatives.

Hueckel [7] proposed a method for edge-fitting and edge detection and later it has been simplified by Rosenfeld [17]. Though the gradient based operators are computationally simple and amenable to the hardware implementation, the Laplacian operator, which is a second order derivative is mostly used to establish the location of edges present in the image. This operator cannot be used in its natural form for edge detection.

Instead, it is applied with the Gaussian function which is known as Laplacian of Gaussian (LoG) function [11].

Goshtasby et al., [5] proposed a curve-fitting approach for edge detection in which the parametric curves are being used to represent the edge contours and fitted to the high-gradient image pixels with its weight proportional to the gradient magnitude of the pixels.

Krishnamoorthi et al., [8] used the curve fitting model based on zero-crossing in the second order derivatives and claimed that the technique is superior to the vector order statistics and entropy schemes in color images. Sarkar et al., [18] reported that the edge detectors are based on optimality criterion.

Canny's [1] operator is very much popular among all the edge detectors and it uses Gaussian filter for smoothing the image before extracting the edges.

Ding et al., [Ding01] have proved that the Canny's algorithm cannot capture the branching edges and edge junctions, but it captures the spurious edges.

Grimson et al., [6] introduced a residual analysis approach for computer vision applications and suggested the zero-crossings of the residual results in locating the edges.

Subsequently, [9] have analysed and reported the differences between the original and the smoothed version, which are the important feature detectors to extract edges in gray scale images [10].

Chen et al., [2] considered the distribution of residuals and found auto correlation for the residual by which they could extract the edge features.

Zheng et al., [24] proposed a hybrid edge detector with the combination of gradient and zero-crossing based on Least Square Support Vector Machine (LS-SVM) with the Gaussian filter. It is reported that it takes lesser time than the Canny's detector with similar performance on edge extraction. In the earlier works, the threshold is chosen on heuristic basis for edge detection. Even in the Canny's edge detector the default value of the upper limit is suggested to be 75th percentile of the gradient strength.

Rakesh et al., [15] reported that this problem has been overcome in their work, but their algorithm needs initial threshold value and parameters as an input. Several methods have been proposed for edge detection with marking points in a digital image. Most of the algorithms typically convolute a filter operator, and then map overlapping input image regions to output signals which lead to a considerable loss in edge detection. Even Gaussian filter suffers with some problems such as edge displacement, vanishing edges and false edges.

Shashank Mathur et. al., [19] have proposed an application model based on fuzzy logic that helps to deal with problems with imprecise and vague information.

Dong-Su Kim et al. [4] proposed a procedure to determine the edge magnitude and direction which are found in the (3 × 3) ideal binary pattern for a block. The average value is calculated for the pixels in a (3 × 3) block, and compared with each pixel in the block. If the pixel is greater than the average value then it is marked as 1, otherwise it is marked as 0. Then, they proposed fixed weight such as (1, 2, 4, 8, 16, 32, 64 and 128) for the 8-neigbouring pixels depending on their zonal position related to the centre pixel and formulated an 8-bit code by adding the weight of the pixels which are marked as 1. This is not justifiable because the pixel values in a (3 × 3) block generally influence the centre pixel more or less equally. In this approach, they used the weight 1 to the pixel in the location (1, 1) and 128 to the pixel in the location (3, 3).

3. EXISTING METHODOLOGY

3.1 LOCAL TETRA PATTERNS

Local Tetra Patterns for the application of content-based image retrieval. This method encodes the spatial relationship between the referenced pixel and its neighbors, which is based on the first order derivatives, along vertical and horizontal directions. The 8 bit tetra pattern for each center pixel is formulated, then separate all patterns into four parts based on the direction of center pixel. Finally, the tetra patterns for each direction are converted into three binary patterns. Similarly, the other tetra patterns for remaining three directions (2, 3 and 4) are converted to 12 binary patterns.

3.2 LOCAL TERNARY PATTERNS

Conventional LBP is extended to a three – valued code called as LTP. It preserves more textural information than LBP. This descriptor perceives the number of transitions or discontinuities in the circular presentation of the patterns. When such transitions are found to follow a rhythmic pattern, they are recorded as uniform LTP.

3.3 PROPOSED METHODOLOGY

The feature extraction is an important process to make efficient image indexing and retrieval. Hence an enhanced technique is used to extract the feature and represent image as a template for image recognition and retrieval.

3.3.1 ORIENTATIONAL FEATURE

There are various models and methods are available for image indexing and retrieval but not sufficient for providing accuracy in recognition process. To develop an image indexing and retrieval system features extracted based on orientation and autocorrelation features of an image. In order to improve the efficiency of the image retrieval system using low-level salient features embedded in the edges are extracted.

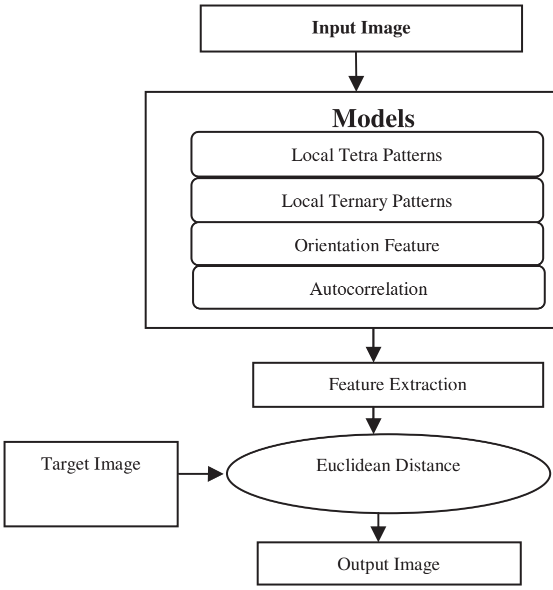


Fig.1 Process flow

In general, matching of texture based image is carried out with the similarity between the areas of the images with similar texture. The technique that has been used for measuring texture similarity is done by calculating the relative brightness of selected pairs of pixels from each image. From these it is possible to measure texture features of the image such as the degree of contrast, coarseness, directionality and regularity, or periodicity, directionality and randomness. Texture queries can be formulated in a similar manner to color queries, by selecting examples of desired textures from a palette or by supplying a query image. The system then retrieves images with texture measures which are most similar to the query.

If the primitives are large, the function decreases slowly with increasing distance where as it decreases rapidly if texture consists of small primitives. The auto-correlation function is defined as:

$$A(r,c) = \frac{1}{(M-r)(M-c)} \frac{\sum_{i=0}^{M-r-1}\sum_{j=0}^{N-c-1} g^0(i,j)\, g^0(i+r,j+c)}{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} g^0(i,j)\, g^0(i,j)} \quad \text{---(3.1)}$$

Here g^0 represents the zero-mean image of the gray level image g(r,c) of size M X N and computed by the equation [g^0=g(r,c)-mean of g]. Nature of function value reveals the presence of texture pattern and its periodicity of representation. However, if the primitives are periodic, then the autocorrelation increases and decreases periodically with distance. The set of autocorrelation coefficients [G_{cff}] are modeled to extract the texture features as below:

$$G_{cff} = \frac{MN \sum_{i=1}^{M-P} \sum_{j=1}^{N-q} g(i,j)\, g(i+p, j+q)}{(M-p)(N-q) \sum_{i=1}^{M} \sum_{j=1}^{N} g^2(i,j)} \quad \ldots (3.2)$$

Here the position difference in the i, j direction is denoted by p and q respectively and their value ranges from (0,0) to (r-1,s-1) such that $1 \leq r \leq M/2$ and $1 \leq s \leq N/2$.The gray scale image with M X N dimension is considered for computation.

When the positional difference (p,q) varies from (0,0) to (r-1,s-1) the extracted texture feature is represented as a matrix as shown below:

$$F_{cff}(p,q) = (G_{ff}(r,s))^{m,n}_{\substack{r=0..m-1 \\ s=0..n-1}} \quad \ldots\ldots\ldots (3.3)$$

In order to represent the identified micro textured regions, the autocorrelation coefficients are calculated using the equation 3.2 and are stored in an array for various sub-images. The computed autocorrelation values range from 0 to 1. A simple transformation (P*100) is applied on the autocorrelation values to obtain decimal number that range from 0 to 100, where P is the autocorrelation coefficient.

3.3.2 AUTOCORRELATION FUNCTION

In statistics, the autocorrelation function of a random process describes the correlation between the processes at different points in time. Let X_t be the value of the process at time t (where t may be an integer for a discrete time process or a real number for a continuous-time process). If X_t has Mean μ and variance σ^2 then the definition of the ACF is

$$R(t,s) = \frac{E[(X_t - \mu)(X_s - \mu)]}{\sigma^2} \quad \ldots (3.4)$$

The expression is not well-defined for all time series. Since the variance σ^2 may be zero (for a constant process) or infinite. If the function R is well defined its value must lie in the range [-1,1], with 1 indicating perfect correlation and -1 indicating perfect anti-correlation.

If X_t is second-order stationary then the ACF depends only on the difference between t and s can be expressed as a function of a single variable. This gives

$$R(k) = \frac{E[(X_i - \mu)(X_{i+k} - \mu)]}{\sigma^2} \quad \ldots (3.5)$$

Where k is the tag |(t-s)|. It is common practice in many disciplines to drop the normalization by σ^2 and use the term autocorrelation interchangeably with auto-covariance.

For a discrete time series of length n{X_1, X_2, ... X_n} with known Mean and Variance, an estimate of the autocorrelation may be obtained as

$$\hat{R}(k) = \frac{1}{(n-k)\sigma^2} \sum_{t=1}^{n-k} [X_t - \mu][X_{t+k} - \mu] \quad \ldots (3.6)$$

for any positive integer k<n. When the true Mean μ is known, this estimate is unbiased. However, if the true Mean and variance of the process are not known and μ and σ^2 replaced by the standard formulae. For sample mean and sample variance, then this estimate is biased. An alternative way a period gram based estimate replaces n-k in the above formula with n. This estimate is always biased it usually has a smaller Mean square error. In image processing, the above definition is often used without the normalization without subtracting the Mean and dividing by the variance. When the autocorrelation function is normalized by mean and variance, it is sometimes referred to as the autocorrelation coefficient. Multi-dimensional autocorrelation is defined similarly.

3.3.3 HORIZONTAL DIRECTIONALITY

In the proposed model, the texture horizontal directionality of each block of size (n x n) is computed by applying q = 0 in the equation (3.2). The auto correlation coefficients on horizontal directionality is modeled and converted into autonums and are represented as

$$F_{hff}(p,0) = (G_{ff}(r,0))^{m}_{r=0..m-1} \quad \ldots (3.7)$$

3.3.4 VERTICAL DIRECTIONALITY

In the proposed model, the texture vertical directionality of each block of size (n x n) is computed by applying p = 0 in the equation (3.2). The auto correlation coefficients on horizontal directionality is modeled and converted into autonums and are represented as

$$F_{vff}(0,q) = (G_{ff}(0,s))^{n}_{s=0..n-1} \quad \text{---}\ (3.8)$$

3.3.5 RECTANGULAR FEATURES

The rectangular grid of each sub-image is represented in a matrix of size (mxn) as shown in the equation

$$f(x,y) = \begin{cases} 1, & \text{if the presence of shape area within the grid} \\ 0, & \text{otherwise} \end{cases} \quad \ldots (3.9)$$

$$G_{m,n} = (g(m,n))^{m,n}_{\substack{m=0..m-1 \\ n=0..n-1}} \quad \ldots (3.10)$$

The general features such as area, perimeter, centroid and mass are computed for rectangular grid in which it satisfies ¼ of presence of shape and represented by f_{ri} where i=1..k.

3.3.6 GENERATION OF FEATURE SET

The feature set is generated with the orientation image based feature vectors for all the sub images. The features of each blocks of the sub image and are represented as below,

$$F_{ori} = \{F_{cff}, F_{hcff}, F_{vcff}, f_{r1}, \ldots, f_{rk}\} \quad \ldots (3.11)$$

Hence, the obtained orientation features values of each block are represented. Then the feature vectors are obtained with the values as mentioned in equation 3.11. Here, the orientation feature set generation is considered for the recognition and retrieval. The extracted feature set of the block of the target image is obtained.

3.4 PROPOSED ALGORITHM

The entire retrieval procedure with the orientation features is presented as simple algorithms hereunder using MATLAB. In two phases the orientation feature, of the images from the databases, are used. In the algorithm-I, procedure to establish feature set is established for each of the images. In algorithm-II, the image retrieval procedure that retrieves top 'm' images from the IDB corresponding to the target image is presented.

3.4.1 Algorithm – I

// generating feature sets //

Input: Input image size from IDB
Output: Feature database
Begin
Step 1: *Read an image from the image database (IDB) of size M×N.*
Step 2: *Divide the input image into non-overlapping blocks of size.*
Step 3: *Perform procedure ori _ feature ( )*
Step 3: *Perform procedure* auto_ corr ( )
Step 4: *Repeat* ***Step1*** *through* ***Step4*** *for all the images in IDB.*
Step 5: *Establish feature database set*
End

3.4.2 Algorithm-II

//Retrieving top *m* relevant images corresponding to the target image//

Input : Target Image
Output : Resultant Image
Begin
Step 1: *Select the target image of size M×N and divide into blocks.*
Step2: *Repeat* ***Step3*** *and* ***Step 4*** *in algorithm I.*
Step3: *Compute the Euclidean distance between the target image and the image set for matching using the equation 3.13.*
Step 4: *Compute the Precision and Recall using the equation 3.14 and equation 3.15.*
Step 5: *Stop*
End

// procedure auto correlation //

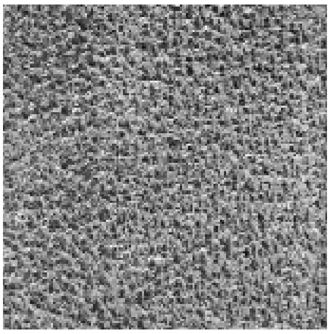
Procedure auto_Corr ()
{
Step 1: *Read an input image from the image database.*
Step 2: *Compute the positional difference by using equation 5.16*
Step3: *Establish feature Matrix with the auto correlation features for all the K sub regions of the input image as discussed in equ.3.2*
Step 4: *Return*
}

// Procedure Orientation Feature//

Procedure ori_feature ( )
Begin
Step 1 : *Identify the closed edges of the input image with rectangular grid.*
Step 2: *Fill the grid values either "1" for presence of rectangular (or) "0" for other.*
Step 3: *Eliminate the shape present in the rectangular grid if the area less than ¼ of the bounded rectangular grid area and count no. of shape (n) for feature extraction.*
Step 4: *Establish the horizontal feature matrix with the auto correlation features of the input image as discussed in section 3.3.2.*
Step 5: *Establish the vertical feature matrix with the auto correlation features of the input image as discussed in section 3.3.4.*
Step 6: *Calculate rectangular based general features such as centroid area, perimeter and mass of selected K^{th} rectangular grid area as discussed in the sub section for 3.3.5.*
Step 7: *Establish the rectangular based general feature set are computed as mentioned in equation 3.10*
Step 8: *Calculate the features set of the input image as mentioned in equation 3.11.*
Step 9: *Repeat* ***Step2*** *through* ***Step6*** *for all the input images and find the feature vectors of each rectangular features.*
Step 10: *Return*
End

4. EXPERIMENTATION & RESULTS

The experimentation is carried out by MATLAB. To validate the effectiveness of the proposed orientation feature based image retrieval system, experimentation is performed with the images in the CORAL image database that contains five hundred 2-D monochrome images of same size. The total images are grouped into 40 classes with 10 images in each class.

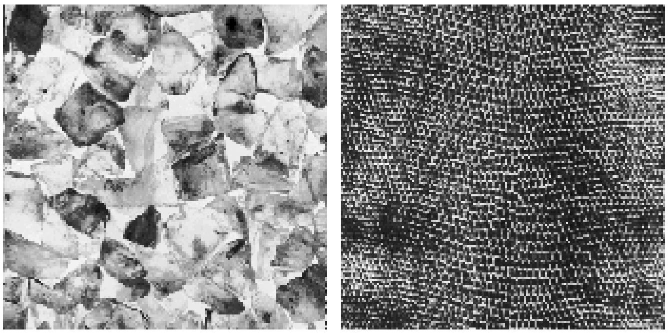


Fig 2.Images Considered for Experimentation

5. SIMILARITY AND PERFORMANCE MEASURES

To find the similarity measures between the images, various metrics are used to measure the distance between features of the images. Some of the well known distance metrics used in for image retrieval is presented below. The Euclidean Distance is calculated as below

$$d_E(x_1, x_2) = \sqrt{\sum_{i=1}^{i=n} \left(x_1(i) - x_2(i)\right)^2} \quad \ldots (3.12)$$

Where $x_1(i)$ is the feature vector of input image i and $x_2(i)$ is the feature vector of the target image i in the image database.

In the texture based image retrieval system Euclidean distance is used to find the distance between the features vectors of the target image I_t and each of the image in the image database (I_i). The difference between two images Ii and It can be expressed as the distance 'd' between the respective feature vectors $Fs(I_i)$ and $Fs(I_t)$.

From the given input image I_i and the target image I_t the Euclidean Distance is calculated as,

$$d_E\left(Fs(I_i), Fs(I_t)\right) = \sqrt{\sum_{i=1}^{i=n} \left(Fs(I_i) - Fs(I_t)\right)^2} \quad \ldots (3.13)$$

Where Fs(Ii) is the feature set of the input image Ii, $Fs(I_t)$ is the n-dimensional feature vector of the target image I_t respectively.

Precision

Precision measures the fraction of retrieved documents that are relevant to a specific query and is analogous to positive predictive value.

$$P(\%) = \frac{\text{No. of relevant images retrieved}}{\text{Total no. of images Retrieved}} \times 100 \quad \text{-- (3.14)}$$

Recall

Recall measures the fraction of all the relevant documents in a collection that are retrieved by a specific query and similar to the concept of sensitivity. Here, recall is the number of figure captions that were indexed by a concept divided by the number of captions in which the concept was actually present.

$$R(\%) = \frac{\text{No. of relevant images retrieved}}{\text{Total no. of relevant images in DB}} \times 100 \quad \text{-- (3.15)}$$

6. PERFORMANCE EVALUATION

The proposed feature extraction is experimented with the images collected from the standard database CORAL consisting of 1000 images are of the size (m x n) as shown in fig.2. From the below Table 1.1 shows that recognition percentage of query images with Proposed Model gives the higher retrieval accuracy of 65.13%. The performance was evaluated using the Euclidean Distance classification by analysis of the values in the table the Proposed model is better for image retrieval.

Table 1.1 Comparison Values

Methods	Recognition
Local Binary Pattern(LBP) [24]	43.62
Local Tetra Patterns (LTP) [24]	49.05
Local Ternary Patterns(LTrPs) [24]	48.79
Proposed Orientation Model	**65.13**

From the below figure shows the pictorial representation of the performance evaluated. By analyzing the obtained results the Proposed Model produced the best results.

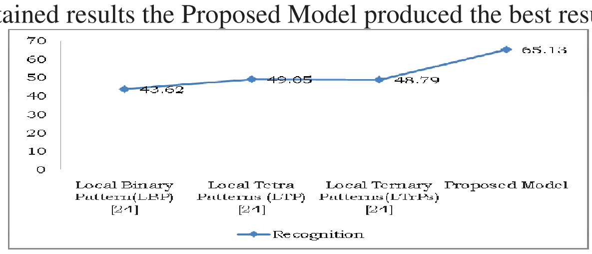


Fig.3 Comparison Graph with Existing Model

Hence, the retrieval rate is estimated in terms of precision and recall. The precision and recall of the proposed method is presented in the following table. The performance of the existing schemes with precision and recall are also obtained. For every comparison, they are also incorporated in the same Table 4.2.

Table 1.2 Comparison Result

Method	Precision	Recall
GLCM [23]	0.72	0.54
Law's Method [23]	0.81	0.68
Proposed Orientation Model	**1.06**	**0.13**

The above Table 1.2 shows the precision and recall of the proposed model in which the proposed model provides the precision 1.06 and recall 0.13 which is better than GLCM and Laws method respectively. Hence, the proposed model is also efficient for image retrieval.

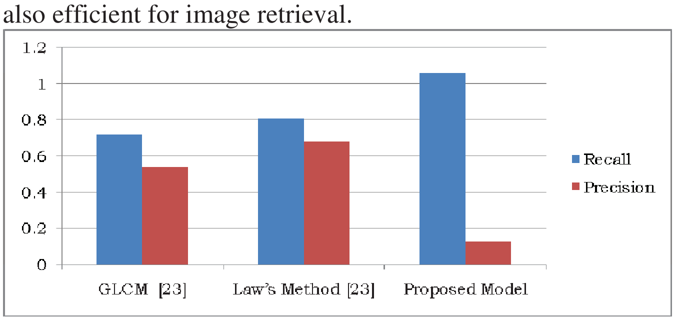


Fig.4 Comparison Graph with Existing Model

The fig.4 shows the pictorial representation of the evaluated performance of the image retrieval. By analyzing the obtained results the Proposed Model produces better results.

7. CONCLUSION

In this paper, integrated model is extended for the edge based image retrieval. The feature sets of the non-textured images are generated with Distance edge orientation histogram sequences of the edge map and are stored in the feature database. The images are then retrieved from the image database with the edge orientation features. In order to achieve the proposed edge based image retrieval are been tested with different image databases. The results are compared with the existing methods.

8. REFERENCES

[1] Canny. J., (1986). "A computational approach to edge detection". IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. 8(6), pp. 679-698.

[2] Chen. M.H., Lee. D., and Pavlidis. T., (1991). "Residual analysis for feature detection", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 13, pp. 30-40.

[3] Ding. L., and Goshtasby. A., (2001). "On the Canny edge detector". Pattern Recognition, vol. 34, pp. 721-725.

[4] Dong-Su Kim., Wang-Heon Lee., and In-So Kweon., (2004). "Automatic edge detection using (3 × 3) ideal binary pixel patterns and fuzzy-based edge thresholding". Pattern Recogntion Letters, vol. 25, pp. 101-106.

[5] Goshtasby. A., and Shyu. H-L., (1995). "Edge detection by curve fitting", Image Vision Comput. vol. 13(3), pp. 169-177.

[6] Grimson. W.E.L., and Pavlidis. T., (1985). "Discontinuity detection for visual surface reconstruction", Computer Vision, Graphics, Image Processing, vol. 30, pp. 316-330.

[7] Hueckel. M., (1971). "An edge operator which locates edge in digital pictures". Journal of ACM. vol. 18, pp. 113-125.

[8] Krishnamoorthi. R., and Bhattacharya. P., (1998). "Color edge extraction using orthogonal polynomials based zero crossing scheme". Information Sciences, vol. 112, pp. 51-65.

[9] Lee. D., Pavlidis T., and Huang. K., (1988). "Edge detection through residual analysis". Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 215-222.

[10] Li. S.Z., (2000). "Roof-Edge Preserving Image Smoothing Based on MRFs". Proceedings of IEEE Transactions on Image Processing, vol. 9(6), pp. 1134-1138.

[11] Marr. D., and Hildreth. E. C., (1980). "Theory of edge detection". Proceedings of Royal Society, vol. 207, pp. 187-217.

[12] Pavlidis. T., and Lee. D., (1988). "Residual analysis for feature extraction", in From Pixel to Features, J.C. Simon, Ed. Amsterdam, Proceedings of Workshop (COST13), pp. 219-227.

[13] Prewitt. J.M.S., (1970). "Object enhancement and extraction". In: B.S. Lipkin and A. Rosenfield, eds., Picture Processing and Psychopictorics, Academic Press.

[14] Qie. P., and Bhandarkar. and S.M., (1996). "An edge detection Technique using local smoothing and statistical hypothesis testing". Pattern Recognition Letters, vol. 17(8), pp. 849-872.

[15] Rakesh. R.R., Probal Chaudhuri., and Murthy. C.A., (2004). "Thresholding in Edge Detection: A Statistical Approach". IEEE Transactions on Image Processing, vol. 13(7), pp. 927-936.

[16] Roberts. L.G.,(1965). "Machine Perception of three dimensional solids". J.T.Tipper Ed., Optical and Electro optical Information Processing.

[17] Rosenfeld. A., (1981). "The Max Roberts operator is Hueckel-type of edge detectors". IEEE Transactions on Pattern Analysis and Machine Intelligence. vol. 3(1), pp. 101-103.

[18] Sarkar. S.Y., and Boyer. K.L., (1991). "Optimal infinite impulse response zero-crossing based edge detectors", Computer Vision Graphics Image Processing: Image Understanding, vol. 54(9), pp. 224-243.

[19] Shashank Mathur., and Anil Ahlawat., (2008). "Application of Fuzzy Logic on Image Edge Detection", Intelligent Technologies and Applications, pp. 24-28.

[20] Shen. J., and Castan. S., (1992). "An optimal linear operator for step edge detection", Graph Models Image Processing, vol. 54(1), pp. 112-133.

[21] Sobel. I.E., (1970). "Camera models and machine perception". Ph.D. Thesis, Stanford University.

[22] Stern. D., and Kurz, L., (1988). "Edge detection in correlated noise using Latin squares models". Pattern Recognition, vol. 21, pp. 119-129.

[23] S. Pannirselvam and R. Krishnamoorthi(2009), "A New Texture Image Retrieval Scheme with Full Range Auto-Regressive (FRAR) Model", International Journal of Soft Computing,pp. 229-235.

[24] Santosh Kumar Vipparthi and Shyam Krishna Nagar (2014),"Color Directional Local Quinary Patterns for Content Based Indexing and Retrieval", Human-centric Computing and Information Sciences,pp.2-13.