CBIR Using Local and Global Properties of Image Sub-blocks

E. R. Vimina[1] and K. Poulose Jacob[2]

[1]*Department of Computer Science, Rajagiri College of Social Sciences, Kerala, India*
[2]*Department of Computer Science, CUSAT, Kerala, India*
vimina_er@yahoo.com, kpj@cusat.ac.in

Abstract

This paper proposes a content based image retrieval (CBIR) system using the local colour and texture features of selected image sub-blocks and global colour and shape features of the image. The image sub-blocks are roughly identified by segmenting the image into partitions of different configuration, finding the edge density in each partition using edge thresholding, morphological dilation and finding the corner density in each partition. The colour and texture features of the identified regions are computed from the histograms of the quantized HSV colour space and Gray Level Co- occurrence Matrix (GLCM) respectively. A combined colour and texture feature vector is computed for each region. The shape features are computed from the Edge Histogram Descriptor (EHD). Euclidean distance measure is used for computing the distance between the features of the query and target image. Experimental results show that the proposed method provides better retrieving result than retrieval using some of the existing methods.

Keywords: *CBIR, Colour histogram, Euclidean distance, GLCM*

1. Introduction

The volume of digital information is growing at an exponential rate with the steady growth of computer power, increasing access to Internet and declining cost of storage devices. Hence to effectively manage the image information, it is imperative to advance automated image learning techniques. Unlike the traditional method of text-based image retrieval in which the image search is based on textual description associated with the images, Content Based Image Retrieval Systems (CBIR) retrieve image information based on the content of the image. These systems retrieve images that are semantically related to the user's query by extracting visual contents of the image such as colour, texture, shape or any other information that can be automatically extracted from the image itself and using it as a criterion to retrieve content related images from the database. The retrieved images are then ranked according to the relevance between the query image and images in the database in proportion to a similarity measure calculated from the features [1, 2, 3].

2. Related Work

Many early CBIR systems perform retrieval based on the global features of the query image [4, 5, 6, 15]. Such systems are likely to fail as the global features cannot sufficiently capture the important properties of individual objects. Recently, much research has focused on region-based techniques [2, 3, 7, 16, 19]. Such systems either subdivide the image into fixed blocks [19, 20, 21] or partition the image into different meaningful regions using segmentation algorithms [2, 3, 7, 23]. Performance of segmentation based methods depends highly on the quality of the segmentation as the average features of all pixels in a segment are often used as the features of that segment. Small areas of incorrect segmentation might make the representation very different from that of the real object. Incorrect segmentation may also affect the shape features. Also accurate segmentation is still a challenging problem and the computational load of segmentation method is heavier. Other CBIR systems [16, 30] extract salient points (also known as interest points) [28, 29], which are locations in an image where there is a significant variation with respect to a chosen image feature. In salient point based methods, feature vector is created for each salient point and the selection of the number of salient points is very important. These representations enable a retrieval method to have a representation of different local regions of the image, and thus these images can be searched based on their local characteristics.

3. Proposed Method

In the proposed method fixed block segmentation method is used as pixel-wise segmentation is computationally costly and accurate segmentation is still a challenging problem. Here the images are divided into different sized blocks for feature extraction. Feature vectors are extracted from selected grids of four different configurations (3x3 grid, horizontal partitions, vertical partitions and central block) and the entire image (Figure 1). Unlike some block based image retrieval systems that uses all the sub-blocks for feature extraction and similarity measurement, our system uses selected blocks only reducing the computational time and cost.

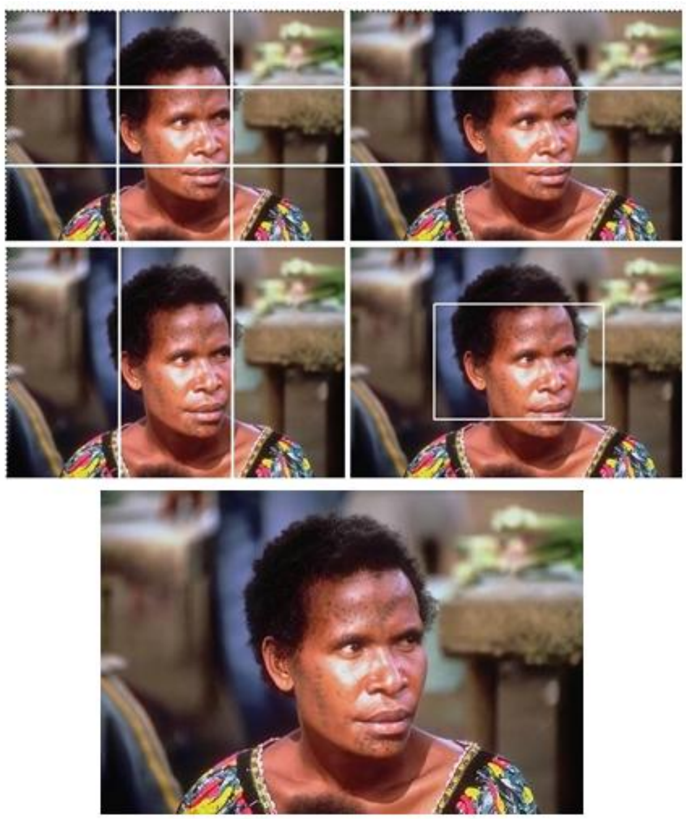

Figure 1. Different Image Configurations for Feature Extraction

To identify the grids /object regions, first the grayscale image is computed image and edge map is detected using Sobel edge filter with a threshold value of τ (τ <1 so that the edges are boosted). The gaps in the edge map are bridged by dilating it with 'line' structuring element, that consists of three 'on' pixels in a row, in the 0, 45, 90 and 135 directions. The holes in the resultant image are then filled to get the approximate location of the objects. The objects are identified correctly if the background is uniform.

A sub-block is selected for further processing, feature extraction and is identified as region of interest (ROI) if τ'% of the sub-block is part of the object region. Ie, if the number of white pixels in that sub-block is τ'% of the sub-block with maximum white pixel density, it is identified as a region of interest. Here we have taken τ'=50%. For example, for the 3x3 partitioned image in Figure 2, regions 2, 3, 5, 7, 8 and 9 are the ROIs. Only these sub-blocks take part in further computations for calculating the similarity along with the global colour and shape features of the entire image [26].

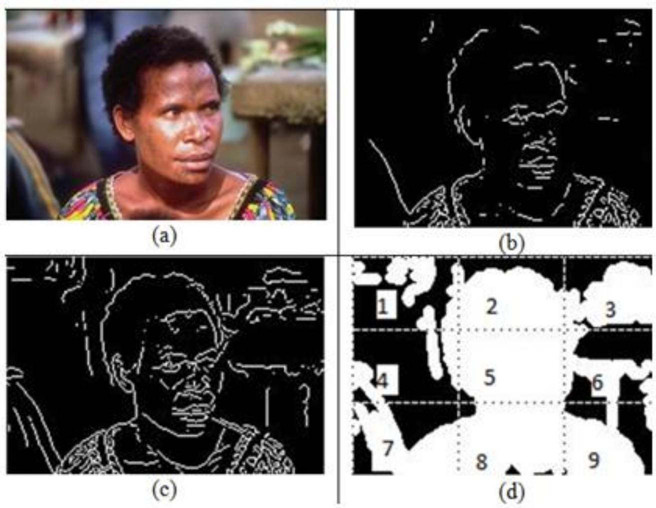

Figure 2. (a)Original Image; (b)Edge Map after Sobel Edge Filtering; (c) Edge Map after Edge Thresholding; (d) Region Identification

For the horizontal and vertical sub blocks also, only blocks with white pixel density greater than a particular threshold only is taken as ROI and these sub-blocks only take part in further similarity computations.

Figure 3. Detected Corners

After identifying the ROIs, the corner density of each identified ROI is determined (Fig.3) using the corner detection algorithm based on curvature scale space [12]. Corners represent the point where two edges meet and the human eye is more sensitive to changes made in these places. The identified ROIs are then arranged in the descending order of corner density in them assuming that the ROIs with greater number of corners provide more distinctive information than the ones with lesser number of corners and for the comparison purpose according to the minimum distance algorithm specified in Section 5.

4. Feature Extraction

After identifying the image sub-blocks/ prominent regions of object, colour and texture features for each region are computed.

4.1 Colour

Colour is one of the most effective, simplest and widely used low level visual features employed in CBIR. Among the proposed approaches based on colour, color histogram is employed extensively [17, 18]. The color histogram is obtained by counting the number of times each color occurs in the image array. Color histogram H for a given image is defined as a vector

$$H=\{H[0], H[1], H[2],\ldots,H[I],\ldots.H[N]\} \quad (1)$$

where i represent the color in color histogram and H[i] represent the number of pixels of color i in the image, and N is the number of bins used in color histogram. For comparing the histogram of different sizes, color histogram should be normalized. The normalized color histogram is given as

$$H'=H/p \quad (2)$$

where, p is the total number of pixels in the image [7].

As some colour spaces (LUV, HSV) coincide better with human perception than the basic RGB colour space that is normally used in monitors, we use HSV colour space for extracting the colour features. The HSV space is quantized to 18 bins for hue, 3 bins for saturation and 3 bins for value. The histogram of each of these channels are extracted resulting in a 24 dimensional colour feature vector that is normalized in the range of [0,1]. For each image both global and local colour features are extracted.

4.2 Texture

Texture can be considered as repeating patterns of local variation of pixel intensities. Unlike colour, texture occurs in a region than at a point. A number of techniques have been used for measuring the texture features such as Gabor filter [24], fractals [25], wavelets, co-occurrence matrix etc. Using these texture features like contrast, coarseness, directionality and regularity can be measured. The Gray Level Co-occurrence Matrix (GLCM) is a statistical method of examining texture that considers the spatial relationship of pixels [8]. It is a matrix showing how often a pixel with the intensity (gray-level) value i occurs in a specific spatial relationship to a pixel with the value j. It is defined by $P(i, j| d,\theta)$, which expresses the probability of the couple of pixels at θ direction and d interval. Once the GLCM is created various features can be computed from it. The most commonly used features are contrast, energy, entropy, correlation and homogeneity (Table 1). We have taken d=1 and $\Theta = 0^o, 45^o, 90^o$ and 135^o for computing the texture features. Contrast, energy, correlation and homogeneity are taken in all the four directions and entropy of the whole block is separately calculated as it gave better retrieving results. Thus 17 texture feature vectors are calculated for each sub-block.

Table 1. Texture Features

Sl. No	Texture Features	
	Feature	*Formula*
1	Contrast	$\sum_{i,j} \lvert i-j \rvert^2 p(i,j)$
2	Correlation	$\sum_{i,j} \frac{(i-\mu i)(j-\mu j)p(i,j)}{\sigma_i \sigma_j}$
3	Energy	$\sum_{i,j} p(i,j)^2$
4	Entropy	$-\sum_{i,j} (p(i,j).*\log(p(i,j))$
5	Homogeneity	$\sum_{i,j} \frac{p(i,j)}{1+\lvert i-j \rvert}$

4.3 Shape

The shape features are extracted using the edge histogram descriptor (EHD). It represents the local edge distribution by dividing image space into 4×4 subimages and representing the local distribution of each subimage by a histogram. The fact that the EHD consists of the local-edge histograms only, makes it very flexible. To generate the histogram, edges in the sub-images are categorized into five types; vertical, horizontal, 45-degree diagonal, 135-degree diagonal and non-directional edges (Figure 4). Since there are 16 sub-images, a total of 5×16 =80 histogram bins are required [13, 14]. Each subimage is further divided into nonoverlapping square image blocks with particular size which depends on the image resolution. Each of the image blocks is then classified into one of the five mentioned edge categories or as a nonedge block. A simple method to do this classification is to treat each image-block as a 2× 2 super-pixel image-block and apply appropriate oriented edge detectors (Figure 5) to compute the corresponding edge strengths. The edge detector with maximum edge strength is then identified. If this edge strength is above a given threshold, then the corresponding edge orientation is associated with the image-block. If the maximum of the edge strengths is below the given threshold, then that block is not classified as an edge block.

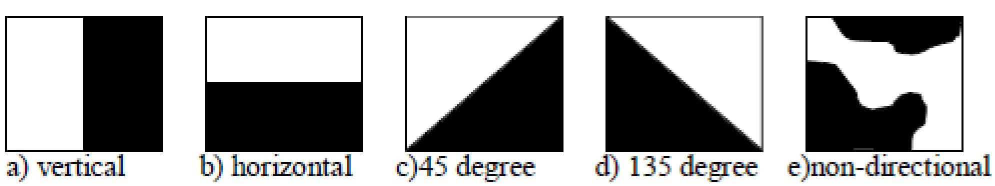

Figure 4. Five Types of Edges in the Edge Histogram Descriptor

$$\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \quad \begin{bmatrix} \sqrt{2} & 0 \\ 0 & -\sqrt{2} \end{bmatrix} \quad \begin{bmatrix} 0 & \sqrt{2} \\ -\sqrt{2} & 0 \end{bmatrix} \quad \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix}$$

Figure 5. Filter Coefficients for Edge Detection

5. Similarity Measure

Euclidean distance is used for computing the similarity between the given pair of images. It is given by,

$$d_{(I1,I2)} = \sqrt{(F_{I1}-F_{I2})^2} \tag{3}$$

Where, F_{I1} and F_{I2} are the feature vectors of image I_1 and I_2.

5.1 Minimum Distance Algorithm

For computing the minimum distance between the regions of the images the algorithm in [26] is used which is described below.

For each ROI in the query image, the colour and texture features are computed and is compared with each ROIs of the target images (Figure 6). Assume that image I_1 has *m* ROIs represented by $R_1 = \{r_1, r_2, \ldots\ldots, r_m\}$ and I_2 has *n* ROIs represented by $R_2 = \{ r'_1, r'_2, \ldots\ldots r'_n\}$. The ROIs are arranged in descending order of corner density in them. This means that the ROIs r_1, r'_1 will be having the maximum number of corners in I_1 and I_2 respectively. Let the distance between r_i and r'_j be d(ri,r'j) denoted as $d_{i,j}$. Every region r_i of R_1 is compared with same number of regions r_j of R_2. This results in '*m*' comparisons for a single region in R_1 and *m* distance measures. These distances are stored in ascending order in an array and the minimum distance only is taken for the final computation of the distance D; the distance between I_1 and I_2. Thus out of the $m \times m$ distances *m* lowest distances are added to get the distance D. This means that if image I_1 is compared with itself, D will be equal to zero. The algorithm for computing the minimum distance between two images is described below:

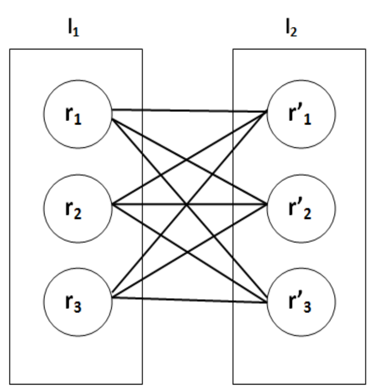

Figure 6. m Regions of I1 are Compared with First m Regions of I2

Input: R_1, R_2; the ROIs of the query and the target image
Output:D, minimum distance between regions of I_1 and I_2
begin
  for each region in the query image I_1, i=1 to *m* **do**
    for first *m* regions in the target image I_2, j=1 to *m* **do**
        compute distance d[j]=d i,j;
    end
    Sort distance array 'd' in ascending order;
    D=D+d[1];
  end
end

'*d*' is the array containing the distances between the r_i of R_1 with the *m* regions of R_2. Similarly the minimum distance between the horizontal and vertical blocks are also computed as D_h and D_v respectively. The final distance between I_1 and I_2 is given by

$$D' = D + D_h + D_v + d_{global\_colour\_feature} + d_{central\ block\_colour\_texture\_feature} + d_{global\_shape\_feature} \quad (4)$$

where, $d_{global\_colour\_feature}$ and $d_{global\_shape\_feature}$ are the Euclidean distance between the global colour and shape feature vectors of I1 and I2 and $d_{central\ block\_colour\_texture\_feature}$ is the distance between the feature vectors of the central blocks of I1 and I2 .

6. Experimental Results

The Wang's image database [9] of 1000 images, which is considered to be one of the benchmark databases for CBIR, consisting of 10 categories is used for evaluating the performance of the proposed method. Each category contains 100 images. A retrieved image is considered to be correct if and only if it is in the same category as the query. For each query, a preselected number of images are retrieved which are illustrated and listed in the ascending order of the distance between the query and the retrieved images. The results of the proposed method is compared with that of [10, 11, 16] and [27] in terms of average precision. Precision (P) of retrieved results is given by

$$P(k) = n_k/k \quad (5)$$

Where, k is the number of retrieved images, n_k is the number of relevant images in the retrieved images. The average precision of the images belonging to the q^{th} category A_q is given by

$$\overline{P}_q = \sum_{k \in A_q} P(I_k) \Big/ \left|A_q\right|,\ q = 1, 2, \ldots .10. \quad (6)$$

The final average precision is

$$\overline{P} = \sum_{q=1}^{10} \overline{P}_q \Big/ 10 \quad (7)$$

Table 2 shows the average precision of the retrieved images for different categories when k=20 for different methods. It is seen that for most of the categories the proposed method provides better or comparable results with that of the other methods. For a few categories like 'Beaches' and 'Mountains' the performance of the proposed method is lower than that of some of the compared methods because of the similarity of the background of the images. For the categories 'Dinosaur' and 'Flowers' the average precision when k=20 is very high. This means that for images with single object the proposed algorithm works better than the compared algorithms.

Table 2. % Average Precision (K=20) of Retrieved Images using different Methods

Category	% Average precision of retrieved images for k=20				
	Jhanwar et al[11]	Hung and Dai's [10]	Minakshi Banerjee et al [16]	CTDCIRS [27]	Proposed method
Africa	45.25	42.40	60.25	56.20	71.56
Beaches	39.75	44.55	55.23	53.60	46.15
Buildings	37.35	41.05	60.5	61.00	56.20
Bus	74.10	85.15	70.59	89.30	87.40
Dinosaur	91.45	58.65	95.00	98.40	99.95
Elephant	30.40	42.55	75.50	57.80	58.45
Flowers	85.15	89.75	80.50	89.90	95.15
Horse	56.80	58.90	90.00	78.00	92.65
Mountains	29.25	28.5	65.80	51.20	35.65
Food	36.95	42.65	55.80	69.40	67.35
Average	**52.64**	**53.24**	**70.91**	**70.48**	**71.05**

Figure 6 depicts the top 19 retrieved images for two sample query image using proposed method and global HSV histogram+ GLCM texture+ global shape based retrieval. In each set, on top left corner is the query image and the retrieved images are listed according to their distance with the query image. (a) shows the retrieved results using global features and (b) shows that using the proposed method .

(a)

(b)

Figure 6. Retrieved Images: (a) Global Features based Method; (b) Proposed Method

7. Conclusion and Future Work

A content based image retrieval system using the colour and texture features of selected sub-blocks/ automatically extracted object regions and global colour and shape features of the image is proposed. The colour features are extracted from the histograms of the quantized HSV color space, texture features from GLCM and shape features from EHD. Euclidean distance measure is used for computing similarity. Unlike the most sub-block based methods that involves all the sub-blocks of the query image to be compared with that of the candidate images, our system involves only selected sub-blocks for similarity measurement, thus reducing the number of comparisons and computational cost. Experimental results also show that the proposed method provides better retrieving result than some of the existing methods. Future work aims at generation of fuzzy rules that may improve the retrieval precision. Also the proposed method has to be tested on various databases to test the robustness.

References

[1] J. Li and J. Z. Wang, "Real-time computerized annotation of pictures", Proceedings of the 14th annual ACM international conference on Multimedia, **(2006)**, pp. 911-920.

[2] Y. Chen, J. Z. Wang and R. Krovetz, "CLUE: Cluster-based retrieval of images by unsupervised learning", IEEE Transactions on Image Processing, vol. 14, no. 8, **(2005)**, pp. 1187-1201.

[3] J. Z. Wang, J. Li and G. Wiederhold, "SIMPLIcity: Semantics-sensitive Integrated Matching for Picture Libraries", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 9, **(2001)**, pp. 947-963.

[4] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang and B. Dom, "Query by Image and Video Content: The QBIC System", IEEE Computer, vol. 28, **(1995)**, pp. 23-32.

[5] A. Pentland, R. Picard and S. Sclaroff, "Photobook: Content-based Manipulation of Image Databases", Proc. SPIE Storage and Retrieval for Image and Video Databases II, SanJose, CA, **(1994)**, pp. 34–47.

[6] M. Stricker and M. Orengo, "Similarity of Color Images", in Proc. SPIE Storage and Retrieval for Image and Video Databases, **(1995)**, pp. 381-392.

[7] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein and J. Malik, “Blobworld: A System for Region-Based Image Indexing and Retrieval”, Proc. Visual Information Systems, **(1999)**, pp. 509-516.

[8] R. M. Haralick, K. Shanmugan and I. Dinstein, "Textural Features for Image Classification", IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-3, **(1973)**, pp. 610-621.

[9] http://wang.ist.psu.edu/docs/related/.

[10] P. W. Huang and S. K. Dai, “Image retrieval by texture similarity”, Pattern Recognit., vol. 36, no. 3, **(2003)**, pp. 665–679.

[11] N. Jhanwar, S. Chaudhuri, G. Seetharaman and B. Zavidoviqu, “Content based image retrieval using motif co-occurrence matrix”, Image and Vision. Computing, vol. 22, no. 14, **(2004)**, pp. 1211-1220.

[12] X. C. He and N. H. C. Yung, “Curvature Scale Space Corner Detector with Adaptive Threshold and Dynamic Region of Support”, Proceedings of the 17th International Conference on Pattern Recognition, vol. 2, **(2004)**, pp. 791-794.

[13] “Overview of the MPEG-7 standard”, **(2001)** December, ISO/IEC/TC/SC29/WG11 N3914.

[14] B. S. Manjunath, P. Salembier and T. Sikora, "Introduction to MPEG-7", John Willey & Sons, Ltd., **(2002)**, pp. 183-184.

[15] S. Murala, A. B. Gonde and R. P. Maheshwari, “Color and Texture Features for Image Indexing and Retrieval”, 2009 IEEE International Advance Computing Conference, (**2009**), , March, Patiala, India, pp. 1411-1416.

[16] B. Minakshi, K. Malay and M. Pradipta, “Content-based image retrieval using visually significant point features”, Fuzzy Sets & Systems, vol. 160, no. 23, **(2009)**, pp. 3323-3341.

[17] M. J. Swain and D. H. Ballard, “Color indexing”, Int. Journal of Computer Vision, **(1991)**, pp. 11-32.

[18] J. Hafner, H. S. Sawhney and W. Wquitz, “Efficient color histogram indexing for quadratic form distance functions”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 7, **(1995)**, pp. 729–736.

[19] M. J. Hsiao, J. P. Huang, T. Tsai and T. W. Chiang, “An Efficient and Flexible Matching Strategy for Content-based Image Retrieval”, Life Science Journal, vol. 7, no. 1, **(2010)**, pp. 99–106.

[20] Q. Tian, Y. Wu and T. S. Huang, “Combine user defined region-of-interest and spatial layout for image retrieval”, International Conference on Image Processing, vol. 3, **(2000)**, pp. 746 – 749.

[21] P. Howarth and S.Ruger, "Robust texture features for still-image retrieval", IEEE. Proceedings of Visual Image Signal Processing, vol. 152, no. 6, **(2005)**.

[22] R. Datta, D. Joshi, J. Li and J.Z. Wang, "Image Retrieval: Ideas, Influences, and Trends of the New Age", ACM Computing Surveys, vol. 40, **(2008)**, pp. 1-60.

[23] W. T. Su, J. C. Chen and J. J. Lien, “Region Based Image Retrieval System with Heuristic Pre-clustering relevance feedback”, Expert systems with Applications, vol. 37, **(2010)**, pp. 4984-4998.

[24] B. S. Manjunath and W. Y. Ma, “Texture Features for Browsing and Retrieval of Image Data”, IEEE transactions on PAAMI, vol. 18, no. 8. **(1996)**, pp. 837 – 842.

[25] L. M Kaplan, “Fast texture database retrieval using extended fractal features”, SPIE 3312, SRIVD, vol. 1, **(1998)**, pp. 162-173.

[26] E. R. Vimina and K. P. Jacob, “Content Based Image Retrieval Using Low Level Features of Automatically Extracted Regions of Interest”, 4th International Conference on Electronics Computer Technology -ICECT 2012, **(2012)**, Kanyakumari, India, pp. 487-490.

[27] M. B. Rao, B. P. Rao and A. Govardhan, “CTDCIRS: Content based Image Retrieval System based on Dominant Color and Texture Features”, International Journal of Computer Applications, vol. 18, no. 6, **(2011)**, pp. 0975-8887.

[28] D. G. Lowe, “Distinctive image features from scale-invariant keypoints”, International journal of computer vision, vol. 60, **(2004)**, pp. 91-110.

[29] H. Bay, A. Ess, T. Tuytelaars and L. Van Gool, “Speeded-Up Robust Features (SURF)”, Computer Vision and Image Understanding, vol. 110, **(2008)**, pp. 346–359.

[30] P. S. Hiremath, J. Pujari, “Content Based Image Retrieval using Color Boosted Salient Points and Shape features of an image”, International Journal of Image Processing, vol. 2, no. 1, **(2008)**, pp. 10-17.

Authors

E. R. Vimina, Department of Computer Science, Rajagiri College of Social Sciences, India received B.Tech in Electrical and Electronics Engineering from Mahatma Gandhi University, India and ME degree in Computer Science and Engineering from Bharathiar University, India. She is currently pursuing doctoral research in Image Retrieval at Cochin University of Science and Technology, India. Her research interests include Image Retrieval and Artificial intelligence.

Dr. K. Poulose Jacob, Professor of Computer Science at Cochin University of Science and Technology since 1994, is currently Director of the School of Computer Science Studies. His studies and research in Multi-microprocessor Applications earned him his PhD from Cochin University of Science and Technology. He is the founder Director of the School of Computer Science Studies. He has been Director of the CUSAT Computer Centre and had initiated the Centre for Information Resource Management. He has been the Dean of the Faculty of Engineering and is presently Chairman, Board of Studies in Computer Science. He has presented research papers in several International Conferences in Europe, USA, UK, Australia and other countries. He has delivered invited talks at several national and international events. He has been a member of several academic bodies of different Universities and Institutes. He is on the editorial board of international journals in Computer Science. He is a Professional member of the ACM and a Life Member of the Computer Society of India.