Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition Letters 28 (2007) 974–983
Pattern Recognition Letters
www.elsevier.com/locate/patrec

An Integrated Color and Intensity Co-occurrence Matrix

A. Vadivel [a], Shamik Sural [b,*], A.K. Majumdar [a]

[a] *Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur 721302, India*
[b] *School of Information Technology, Indian Institute of Technology, Kharagpur 721302, India*

Received 29 March 2005; received in revised form 21 September 2006
Available online 23 January 2007

Communicated by R. Manmatha

Abstract

The paper presents a novel approach for representing color and intensity of pixel neighborhoods in an image using a co-occurrence matrix. After analyzing the properties of the HSV color space, suitable weight functions have been suggested for estimating relative contribution of color and gray levels of an image pixel. The suggested weight values for a pixel and its neighbor are used to construct an Integrated Color and Intensity Co-occurrence Matrix (ICICM). We have shown that if the ICICM matrix is used as a feature in an image retrieval application, it is possible to have higher recall and precision compared to other existing methods.

Keywords: Co-occurrence matrix; HSV color space; ICICM; Image retrieval

1. Introduction

Color and texture are two low-level features widely used for image classification, indexing and retrieval. Color is usually represented as a histogram, which is a first order statistical measure that captures global distribution of color in an image (Swain and Ballard, 1991; Gevers and Stokman, 2004). One of the main drawbacks of the histogram-based approaches is that the spatial distribution and local variations in color are ignored. Local spatial variation of pixel intensity is commonly used to capture texture information in an image. Grayscale Co-occurrence Matrix (GCM) is a well-known method for texture extraction in the spatial domain (Haralick et al., 1973). A GCM stores the number of pixel neighborhoods in an image that have a particular grayscale combination. Let I be an image and let p and N_p respectively denote any arbitrary pixel and its neighbor in a given direction. If GL denotes the total number of quantized gray levels and gl denotes the individual gray levels, where, $\mathrm{gl} \in \{0,\ldots,\mathrm{GL}-1\}$, then each component of GCM can be written as follows:

$$\mathrm{gcm}(i,j) = \Pr((\mathrm{gl}_\mathrm{p}, \mathrm{gl}_{N_\mathrm{p}}) = (i,j)) \tag{1}$$

$\mathrm{gcm}(i,j)$ is the number of times the gray level of a pixel p denoted by gl_p equals i, and the gray level of its neighbor N_p denoted by $\mathrm{gl}_{N_\mathrm{p}}$ equals j, as a fraction of the total number of pixels in the image. Thus, it estimates the probability that the gray level of an arbitrary pixel in an image is i, and that of its neighbor is j. One GCM matrix is generated for each possible neighborhood direction, namely, 0°, 45°, 90° and 135°. Average and range of 14 features like Angular Second Moment, Contrast, Correlation, etc., are generated by combining all the four matrices to get a total of 28 features (Haralick et al., 1973). In the GCM approach for texture extraction, color information is completely lost since only pixel gray levels are considered.

To incorporate spatial information along with the color of image pixels, a feature called color correlogram has recently been proposed. It is a three dimensional matrix that represents the probability of finding pixels of any two given colors at a distance '*d*' apart (Huang et al.,

* Corresponding author. Tel.: +91 3222 282330; fax: +91 3222 282206.
E-mail addresses: vadi@cc.iitkgp.ernet.in (A. Vadivel), shamik@cse.iitkgp.ernet.in (S. Sural), akmj@cse.iitkgp.ernet.in (A.K. Majumdar).

0167-8655/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2007.01.004

1997). Auto correlogram is a variation of correlogram, which represents the probability of finding two pixels with the same color at a distance 'd' apart. This approach can effectively represent color distribution in an image. However, correlogram features do not capture intensity variation. Many image databases often contain both color as well as gray scale images. The color correlogram method does not constitute a good descriptor in such databases.

Another method called Color Co-occurrence Matrix (CCM) has been proposed to capture color variation in an image (Shim and Choi, 2003). CCM is represented as a three-dimensional matrix, where color pair of the pixels p and N_p are captured in the first two dimensions of the matrix and the spatial distance 'd' between these two pixels is captured in the third dimension. This approach is a generalization of the color correlogram and reduces to the pure color correlogram for $d = 1$. CCM is generated using only the Hue plane of the HSV (Hue, Saturation and Intensity Value) color space. The Hue axis is quantized into HL number of levels. If individual hue values are denoted by hl, where $\mathrm{hl} \in \{0, \ldots, \mathrm{HL} - 1\}$, then each component of CCM can be written as follows:

$$\mathrm{ccm}(i,j) = \Pr((\mathrm{hl}_p, \mathrm{hl}_{N_p}) = (i,j)) \tag{2}$$

Four matrices representing neighbors at angles 0°, 90°, 180° and 270° are considered. This approach was further extended by separating the diagonal and the non-diagonal components of CCM to generate a Modified Color Co-occurrence Matrix (MCCM). MCCM, thus, may be written as follows:

$$\mathrm{MCCM} = (\mathrm{CCM}_D, \mathrm{CCM}_{ND}) \tag{3}$$

Here, CCM_D and CCM_{ND} correspond to the diagonal and off-diagonal components of CCM. The main drawback of this approach is that, like correlogram, it also captures only color information and intensity information is completely ignored.

An alternative approach is to capture intensity variation as a texture feature from an image and combine it with color features like histograms using suitable weights (Manjunath et al., 2001). One of the challenges of this approach is to determine suitable weights since these are highly application-dependent. In certain applications like Content-based Image Retrieval (CBIR), weights are often estimated from relevance feedback given by users (Aksoy and Haralick, 2000; Wu and Zhang, 2002). While relevance feedback is sometimes effective, it makes the process of image retrieval user-dependent and iterative. There is also no guarantee on the convergence of the weight-learning algorithms. In order to overcome these problems, researchers have tried to combine color and texture features together during extraction.

Palm (2004) proposed two approaches for capturing color and intensity variations from an image using the LUV color space. In the Single-channel Co-occurrence Matrix (SCM), variations for each color channel, namely, L, U and V are considered independently. In the Multi-channel Co-occurrence Matrix (MCM), variations are captured taking two channels at a time – UV, LU and LV. Since the LUV color space separates out chrominance (L and U) from luminance (V), SCM in effect, generates one GCM and two CCMs from each image independently. As a result, correlation between the color channels is lost. However, in MCM, the count of pairwise occurrences of the values of different channels of the color space is captured. Thus, each component of MCM can be written as follows:

$$\mathrm{mcm}_{UV}(i,j) = \Pr((u_p, v_{N_p}) = (i,j)) \tag{4a}$$

$$\mathrm{mcm}_{LU}(i,j) = \Pr((l_p, u_{N_p}) = (i,j)) \tag{4b}$$

$$\mathrm{mcm}_{LV}(i,j) = \Pr((l_p, v_{N_p}) = (i,j)) \tag{4c}$$

Here, $\mathrm{mcm}_{UV}(i,j)$ is the number of times the U chromaticity value of a pixel p denoted by u_p equals i, and the V chromaticity value of its neighbor N_p denoted by v_{N_p} equals j, as a fraction of the total number of pixels in the image. Similarly, $\mathrm{mcm}_{LU}(i,j)$ and $\mathrm{mcm}_{LV}(i,j)$ are defined. One MCM matrix is generated for each of the four neighborhood directions, namely, 0°, 45°, 90° and 135°.

Deng and Manjunath (2001) proposed a two-stage method called JSEG, which combines color and texture after image segmentation. In the first stage, colors are quantized to the required levels for differentiating between various regions of an image. Pixel values of the regions are then replaced by their quantized color levels to form a color map. Spatial variation of color levels between different regions in the map is viewed as a type of texture composition of the image. Yu et al. (2002) suggested the use of color texture moments to represent both color and texture of an image. This approach is based on the calculation of Local Fourier Transformation (LFT) coefficients. Eight templates equivalent to LFT are operated over an image to generate a characteristic map of the image. Each template is a 3×3 filter that considers eight neighbors of the current pixel for LFT calculation. First and second order moments of the characteristic map are then used to generate a set of features.

In this paper, we propose an integrated approach for capturing spatial variation of both color and intensity levels in the neighborhood of each pixel using the HSV color space. In contrast to the other methods, for each pixel and its neighbor, the amount of color and intensity variation between them is estimated using a weight function. Suitable constraints are satisfied while choosing the weight function for effectively relating visual perception of color and the HSV color space properties. The color and intensity variations are represented in a single composite feature known as Integrated Color and Intensity Co-occurrence Matrix (ICICM). While the existing schemes generally treat color and intensity separately, the proposed method provides a composite view to both color and intensity variations in the same feature. The main advantage of using ICICM is that it avoids the use of weights to combine individual color and texture features. We use ICICM feature in an image retrieval application from large image databases. Early result on this work was reported in (Vadivel et al., 2004a). In the next section, we describe the proposed feature extraction technique after introducing some of the properties of the HSV color space. Choice of quantization levels for color and intensity axes, selection of parameter values and a brief overview of the image retrieval application is given in Section 3. Retrieval performance of the proposed scheme with labeled and unlabelled databases is presented in Section 4, and we conclude in the last section of the paper.

2. Integrated color and intensity co-occurrence matrix

We propose to capture color and intensity variation around each pixel in a two-dimensional matrix called Integrated Color and Intensity Co-occurrence Matrix (ICICM). This is a generalization of the Grayscale Co-occurrence Matrix and the Color Co-occurrence Matrix techniques. For each pair of neighboring pixels, we consider their contribution to both color perception as well as gray level perception to the human eye. Some of the useful properties of the HSV color space and their relationship to human color perception are utilized for extracting this feature. In the next sub-section, we briefly explain relevant properties of the HSV color space. In the subsequent subsection, we describe how the properties can be effectively used for generating ICICM.

2.1. Properties of the HSV color space

Sensing of light from an image in the layers of human retina is a complex process with rod cells contributing to scotopic or dim-light vision and cone cells to photopic or bright-light vision (Gonzalez and Woods, 2002). At low-levels of illumination, only the rod cells are excited so that only gray shades are perceived. As the illumination level increases, more and more cone cells are excited, resulting in increased color perception. Various color spaces have been introduced to represent and specify colors in a way suitable for storage, processing or transmission of color information in images. Out of these, HSV is one of the models that separate out the luminance component (Intensity) of a pixel color from its chrominance components (Hue and Saturation). Hue represents pure color, which is perceived when incident light is of sufficient illumination and contains a single wavelength. Saturation gives a measure of the degree by which a pure color is diluted by white light. For light with low illumination, corresponding intensity value in the HSV color space is also low.

The HSV color space can be represented as a hexacone, with the central vertical axis denoting the luminance component, I (often denoted by V for Intensity Value). Hue, is a chrominance component defined as an angle in the range $[0, 2\pi]$ relative to the red axis with red at angle 0, green at $2\pi/3$, blue at $4\pi/3$ and red again at 2π. Saturation, S, is the other chrominance component, measured as a radial distance from the central axis of the hexacone with value between 0 at the center to 1 at the outer surface. For zero saturation, as the intensity is increased, we move from black to white through various shades of gray. On the other hand, for a given intensity and hue, if the saturation is changed from 0 to 1, the perceived color changes from a shade of gray to the most pure form of the color represented by its hue. When saturation is near 0, all the pixels in an image look alike even though their hue values are different. As we increase saturation towards 1, the colors get separated out and are visually perceived as the true colors represented by their hues. Low saturation implies presence of a large number of spectral components in the incident light, causing loss of color information even though the illumination level is sufficiently high. Thus, for low values of saturation or intensity, we can approximate a pixel color by a gray level while for higher saturation and intensity, the pixel color can be approximated by its hue. For low intensities, even for a high saturation, a pixel color is close to its gray value. Similarly, for low saturation even for a high value of intensity, a pixel is perceived as gray. We use these properties to estimate the degree by which a pixel contributes to color perception and gray level perception.

One possible way of capturing color perception of a pixel is to choose suitable thresholds on the intensity and saturation. If the saturation and the intensity are above their respective thresholds, we may consider the pixel to have color dominance; else, it has gray level dominance. However, such a hard thresholding does not properly capture color perception near the threshold values. This is due to the fact that there is no fixed level of illumination above which the cone cells get excited. Instead, there is a gradual transition from scotopic to photopic vision. Similarly, there is no fixed threshold for the saturation of cone cells that leads to loss of chromatic information at higher levels of illumination caused by color dilution. We, therefore, use suitable weights that vary smoothly with saturation and intensity to represent both color and gray scale perception for each pixel.

2.2. Generation of ICICM

ICICM is a two-dimensional matrix which considers relative contribution to color and gray level perception for each pixel p and its neighbor N_p. ICICM consists of four sub matrices and can be represented as follows:

$$\mathrm{ICICM} = \begin{pmatrix} \mathrm{ICICM_{CC}} & \mathrm{ICICM_{CI}} \\ \mathrm{ICICM_{IC}} & \mathrm{ICICM_{II}} \end{pmatrix} \tag{5}$$

Here $\mathrm{ICICM_{CC}}$ represents color perception of the pixel p and color perception of its neighbor N_p. $\mathrm{ICICM_{CI}}$ represents color perception of the pixel p and gray level perception of its neighbor N_p. $\mathrm{ICICM_{IC}}$ and $\mathrm{ICICM_{II}}$ are similarly defined.

Let HL be the number of quantized levels of Hue and GL be the number of quantized levels of Intensity derived from the HSV color space. Each component of $\mathrm{ICICM_{CC}}$, $\mathrm{ICICM_{CI}}$, $\mathrm{ICICM_{IC}}$ and $\mathrm{ICICM_{II}}$ can be written as follows:

$$\mathrm{icicm_{CC}}(i,j)_{|i=0\ldots\mathrm{HL}-1;j=0\ldots\mathrm{HL}-1} = \Pr((\mathrm{hl_p}, \mathrm{hl}_{N_\mathrm{p}}) = (i,j)) \quad (6a)$$

$$\mathrm{icicm_{CI}}(i,j)_{|i=0\ldots\mathrm{HL}-1;j=0\ldots\mathrm{GL}-1} = \Pr((\mathrm{hl_p}, \mathrm{gl}_{N_\mathrm{p}}) = (i,j)) \quad (6b)$$

$$\mathrm{icicm_{IC}}(i,j)_{|i=0\ldots\mathrm{GL}-1;j=0\ldots\mathrm{HL}-1} = \Pr((\mathrm{gl_p}, \mathrm{hl}_{N_\mathrm{p}}) = (i,j)) \quad (6c)$$

$$\mathrm{icicm_{II}}(i,j)_{|i=0\ldots\mathrm{GL}-1;j=0\ldots\mathrm{GL}-1} = \Pr((\mathrm{gl_p}, \mathrm{gl}N_\mathrm{p}) = (i,j)) \quad (6d)$$

Thus, $\mathrm{icicm_{CC}}(i,j)$ is the number of times the color perception of a pixel p denoted by $\mathrm{hl_p}$ equals i, and the color perception of its neighbor N_p denoted by $\mathrm{hl}_{N_\mathrm{p}}$ equals j, as a fraction of the total number of pixels in the image. Similarly, $\mathrm{icicm_{CI}}(i,j)$ is the number of times the color perception of a pixel p denoted by $\mathrm{hl_p}$ equals i, and the gray level perception of its neighbor N_p denoted by $\mathrm{gl}_{N_\mathrm{p}}$ equals j, as a fraction of the total number of pixels in the image.

The dimension of the matrix ICICM is determined by the number of quantization levels HL and GL of the Hue and the Intensity axes, respectively. HL and GL can be computed as follows:

$$\mathrm{HL} = \lfloor 2\pi/Q_\mathrm{H} \rfloor + 1 \quad (7)$$

$$\mathrm{GL} = \lfloor 255/Q_\mathrm{I} \rfloor + 1 \quad (8)$$

Here Q_H and Q_I are the quantization factors for Hue and Intensity, respectively. The dimension of the complete matrix ICICM is $(\mathrm{HL} + \mathrm{GL})^2$. For $Q_\mathrm{H} = 2$ and $Q_\mathrm{I} = 64$, we get $\mathrm{HL} = 4$ and $\mathrm{GL} = 4$, resulting in $\mathrm{DIM_{ICICM}} = 64$.

The ICICM matrix is updated using a weight function $W_\mathrm{col}(S,I)$ that estimates the extent of color perception of a pixel. Based on our observation in Section 2.1, the weight should be a function of both saturation and intensity. Also, considering the preceding observation on the properties of the HSV color space, we introduce the following constraints on the nature of $W_\mathrm{col}(S,I)$:

a. $W_\mathrm{col}(S,I) \in [0,1]$.
b. For $S_1 > S_2$, $W_\mathrm{col}(S_1,I) > W_\mathrm{col}(S_2,I)$.
c. For $I_1 > I_2$, $W_\mathrm{col}(S,I_1) > W_\mathrm{col}(S,I_2)$.
d. $W_\mathrm{col}(S,I)$ changes slowly with S for high values of I.
e. $W_\mathrm{col}(S,I)$ changes sharply with S for low values of I.

Constraints a–c follow directly from the properties of the HSV color space. Constraint d follows from the fact that when intensity is high, the loss of color perception is only due to dilution of color by white light. On the other hand, constraint e is required since for low intensity, loss of color perception is a combined effect of cone cell de-activation and color dilution. While a number of functions may be chosen that satisfy the above relationships, after detailed analysis with large class of images, the following weight function was found to be quite satisfactory (Vadivel et al., 2005).

$$W_\mathrm{col}(S,I) = \begin{cases} S^{r_1*(255/I)^{r_2}} & \text{for } I \neq 0 \\ 0 & \text{for } I = 0 \end{cases} \quad (9)$$

where r_1 and r_2 are constants. The intensity weight of a pixel is computed as a complement of the color weight as given below

$$W_\mathrm{int}(S,I) = 1 - W_\mathrm{col}(S,I) \quad (10)$$

The values of r_1 and r_2 depend on the particular application in which ICICM is used. In a later section (Section 3), we discuss suitable choices of r_1 and r_2 for an image retrieval application.

2.3. Implementation details

The choice of parameters Q_H and Q_I is dependent on the application for which ICICM is used. Any application that requires fine-grained quantization of the hue values, a low Q_H is chosen, resulting in a higher value of HL. Choice of Q_I is also made in a similar manner. A logical view of ICICM considering $Q_\mathrm{H} = 2$, $Q_\mathrm{I} = 64$ and hence, $\mathrm{HL} = 4$, $\mathrm{GL} = 4$, is shown in Fig. 1. In this figure, the four ranges of Hue corresponding to the quantized values 0–3 are $[0,\pi/2)$, $[\pi/2,\pi)$, $[\pi,3\pi/2)$ and $[3\pi/2,2\pi)$. The four ranges have been represented by the symbols P, Q, R and S. The four levels of Intensity (0–3) are represented by K, X, Y and W, where K and W correspond to Black and White. X and Y represent two intermediate levels of gray. For any given combination of current pixel p and neighboring pixel N_p, one component of each sub-matrix is updated. The particular component to be updated in $\mathrm{ICICM_{CC}}$ is the one for which the hue value marked in the lower triangle of a small box in Fig. 1 equals $\mathrm{hl_p}$ and the hue value marked in the upper triangle equals $\mathrm{hl}_{N_\mathrm{p}}$. The component to be updated in $\mathrm{ICICM_{CI}}$ is the one for which the hue value marked in the lower triangle equals $\mathrm{hl_p}$ and the gray level marked in the upper triangle equals $\mathrm{gl}_{N_\mathrm{p}}$. Components to be updated in the other two sub-matrices are chosen in a similar manner.

The quantum of update for the matrix ICICM is determined by the color and the intensity weights of the current pixel p and the neighboring pixel N_p. $\mathrm{ICICM_{CC}}$ entries are

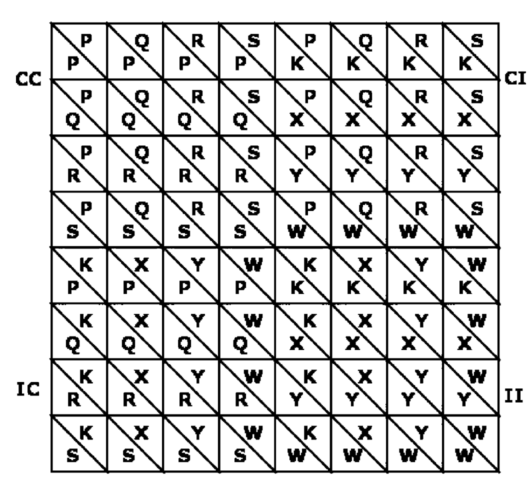

Fig. 1. Logical view of ICICM matrix.

for m = 1 to M // For All rows
for n = 1 to N // For all columns

$$\mathrm{ICICM_{CC}}\left[\frac{hl_{(m,n)}}{Q_H}\right]\left[\frac{hl_{N_{(m,n)}}}{Q_H}\right]=\mathrm{ICICM_{CC}}\left[\frac{hl_{(m,n)}}{Q_H}\right]\left[\frac{hl_{N_{(m,n)}}}{Q_H}\right]+W_{col}^{(m,n)}(S,I)+W_{col}^{N_{(m,n)}}(S,I)$$

$$\mathrm{ICICM_{CI}}\left[\frac{hl_{(m,n)}}{Q_H}\right]\left[\frac{gl_{N_{(m,n)}}}{Q_I}\right]=\mathrm{ICICM_{CI}}\left[\frac{hl_{(m,n)}}{Q_H}\right]\left[\frac{gl_{N_{(m,n)}}}{Q_I}\right]+W_{col}^{(m,n)}(S,I)+W_{\mathrm{int}}^{N_{(m,n)}}(S,I)$$

$$\mathrm{ICICM_{IC}}\left[\frac{gl_{(m,n)}}{Q_I}\right]\left[\frac{hl_{N_{(m,n)}}}{Q_H}\right]=\mathrm{ICICM_{IC}}\left[\frac{gl_{(m,n)}}{Q_I}\right]\left[\frac{hl_{N_{(m,n)}}}{Q_H}\right]+W_{\mathrm{int}}^{(m,n)}(S,I)+W_{col}^{N_{(m,n)}}(S,I)$$

$$\mathrm{ICICM_{II}}\left[\frac{gl_{(m,n)}}{Q_I}\right]\left[\frac{gl_{N_{(m,n)}}}{Q_I}\right]=\mathrm{ICICM_{II}}\left[\frac{gl_{(m,n)}}{Q_I}\right]\left[\frac{gl_{N_{(m,n)}}}{Q_I}\right]+W_{\mathrm{int}}^{(m,n)}(S,I)+W_{\mathrm{int}}^{N_{(m,n)}}(S,I)$$

Fig. 2. Algorithm for updating ICICM.

updated by the sum of the color weight of p and the color weight of N_p. An $\mathrm{ICICM_{CI}}$ entry is updated by the sum of the color weight of p and the intensity weight of N_p. $\mathrm{ICICM_{IC}}$ and $\mathrm{ICICM_{II}}$ entries are updated similarly. An algorithm showing the update process for the complete Integrated Color and Intensity Co-occurrence Matrix is shown in Fig. 2.

It may be observed from the figure that each pixel in the image contributes to an entry in each of the four sub-matrices. The quantum of update in each case is dependent on the color perception and intensity perception of the two pixels. We generate ICICM matrix for four orientations at angles of 0°, 45°, 90° and 135°, respectively. However, the four matrices are updated simultaneously to ensure that the complexity of the algorithm is linear with respect to the size of the image i.e. $O(n)$ – n is the number of pixels in the image. For images of size 350×240, the average time required for ICICM construction on a Pentium IV 1.8 GHz computer running Linux 0.23 s.

At the end of the update process, each ICICM matrix is normalized with respect to the image size so that each component represents the probability of occurrence of a given combination as mentioned in Eqs. (6a)–(6d). The following example illustrates updating of ICICM carried out by a typical pixel.

2.3.1. Example

Consider three choices of H, S and V values for a pixel p and its neighboring pixel N_p shown in Table 1. The entries of each of the sub-matrices $\mathrm{ICICM_{CC}}$, $\mathrm{ICICM_{CI}}$, $\mathrm{ICICM_{IC}}$ and $\mathrm{ICICM_{II}}$ that are updated due to this pixel and the corresponding weights are also shown in the table.

For instance, the third row of the table correspond to $h_p = 2\pi$, $s_p = 0.5$, $v_p = 100$ and $h_{N_p} = 0.4$, $s_{N_p} = 0.9$, $v_{N_p} = 200$. Using Eqs. (7) and (8), $\mathrm{hl_p} = 3$, $\mathrm{gl_p} = 1$, $\mathrm{hl}_{N_p} = 0$ and $\mathrm{gl}_{N_p} = 3$. Using Eqs. (9) and (10), $\mathrm{ICICM_{CC}}[0,,3]$ is thus updated with $W_{\mathrm{col}}^{\mathrm{p}}(S,I) + W_{\mathrm{col}}^{N_p}(S,I) = 0.858 + 0.987 = 1.845$. The other components are similarly updated.

3. Application of ICICM

The Integrated Color and Intensity Co-occurrence Matrix can be used in a number of image processing and pattern recognition problems. We have considered Content-based Image Retrieval applications to study the effectiveness of the ICICM matrix. In the following two sub-sections, we show the effect of quantization levels and the choice of parameters r_1 and r_2 in Eq. (9) on image retrieval performance. We use two standard metrics, namely, recall and precision for measuring performance, which are defined as follows.

$$\mathrm{Recall} = \frac{\text{No. of Relevant Images Retrieved}}{\text{Total No. of Relevant Images}} \tag{11}$$

$$\mathrm{Precision} = \frac{\text{No. of Relevant Images Retrieved}}{\text{Total No. of Images Retrieved}} \tag{12}$$

In the last sub-section, we give an overview of a content-based image retrieval system developed by us.

3.1. Number of quantization levels

To determine the number of components that effectively represents color weight and intensity weight for ICICM, we have used database (MIT Texture database[1]) containing 122 images. The database has 10 image categories each containing approximately 12 images. Any image belonging to the same category as a query image is assumed to be a member of the relevant set. We have considered five different combinations of hue and intensity levels – HL and GL. For example, the combination (6, 2) means that the color weight-color weight will update a (6×6) area, intensity weight-color weight updates a (2×6) area, color weight-intensity weight updates a (6×2) area and intensity weight-intensity weight updates a (2×2) area of ICICM. In Fig. 3, we plot precision vs. recall for different combinations of HL and GL on the test database. It may be noted from the figure that the combination 2×6 gives higher precision compared to the others. The reason is that the database contains a large number of texture-rich images. The corresponding values of Q_H and Q_I of Eqs. (7) and (8) are 5 and 50, respectively.

Other values of the quantization factors result in different numbers of color and gray levels. If the nature of the image database is known a priori, then the values of HL and GL can be appropriately chosen. For image databases with predominantly color images, more number of levels may be assigned to color compared to intensity. On the other hand, for databases with texture rich images, one

[1] http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html

Table 1
ICICM components and weight updates for sample HSV values

HSV (p)	HSV N(p)	ICICM component			
		$ICICM_{CC}$ (i,j, Weight)	$ICICM_{CI}$ (i,j, Weight)	$ICICM_{IC}$ (i,j, Weight)	$ICICM_{II}$ (i,j, Weight)
(0.4, 0.9, 200)	(1.6π, 0.0, 180)	(0, 3, 0.987)	(0, 2, 1.987)	(3, 3, 0.013)	(3, 2, 1.013)
(1.6π, 0.0, 180)	(2π, 0.5, 100)	(3, 3, 0.987)	(3, 1, 0.013)	(2, 3, 1.987)	(2, 1, 1.013)
(2π, 0.5, 100)	(0.4, 0.9, 200)	(3, 0, 1.845)	(3, 3, 0.871)	(1, 0, 1.129)	(1, 3, 0.155)

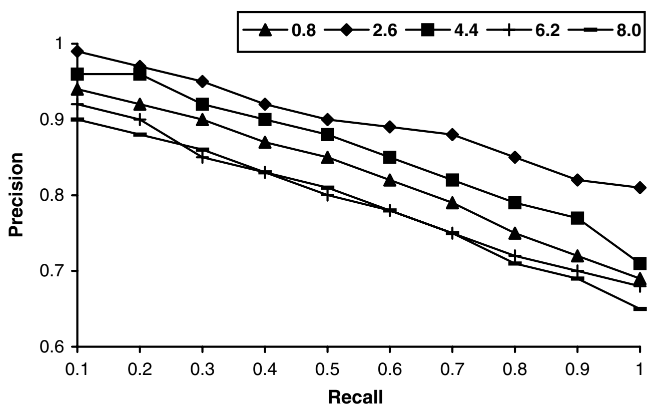

Fig. 3. Precision vs. recall for different combinations of HL and GL.

can choose higher number of gray levels. In our subsequent discussions, we make HL equal to GL, thus giving equal importance to both color and intensity. Further, we chose Q_H and Q_I in Eqs. (7) and (8) such that HL = GL = 4, resulting in a 64-component ICICM matrix. This is done to make the number of feature components comparable to other types of features proposed in the literature.

3.2. Choice of weight function parameters

In order that $W_{col}(S, I)$ of Eq. (9) satisfies conditions (a)–(e), r_1 should take a value slightly higher than 0.0 and r_2 should take a value slightly less than 1.0. To determine the best combination of r_1 and r_2, we have performed another set of experiments on the MIT texture database. We have used different combinations of r_1 and r_2 and calculated recall and precision for 10 nearest neighbors for each query. This was repeated for 25 queries. The average recall and precision values are plotted in Figs. 4a and b. It is seen that a combination of $r_1 = 0.1$ and $r_2 = 0.85$ gives the best average recall and precision. The recall value is higher by at least 5% than the other combinations. The precision is also at least 5% higher than the other combinations. It is further observed that for the same value of r_1, recall and precision vary slowly with r_2. On the other hand, for the same value of r_2, recall and precision vary sharply with r_1. We, therefore, selected $r_1 = 0.1$ and $r_2 = 0.85$ in the proposed CBIR system. A plot of $W_{col}(S, I)$ for different values of S and I is shown in Fig. 5 with $r_1 = 0.1$ and $r_2 = 0.85$. It is seen that the requirements (a)–(e) of Section 2.2 are correctly satisfied by this function.

It is seen from Figs. 3 and 4 that (HL, GL) combinations of (2, 6) and (4, 4) while (r_1, r_2) combinations in the range

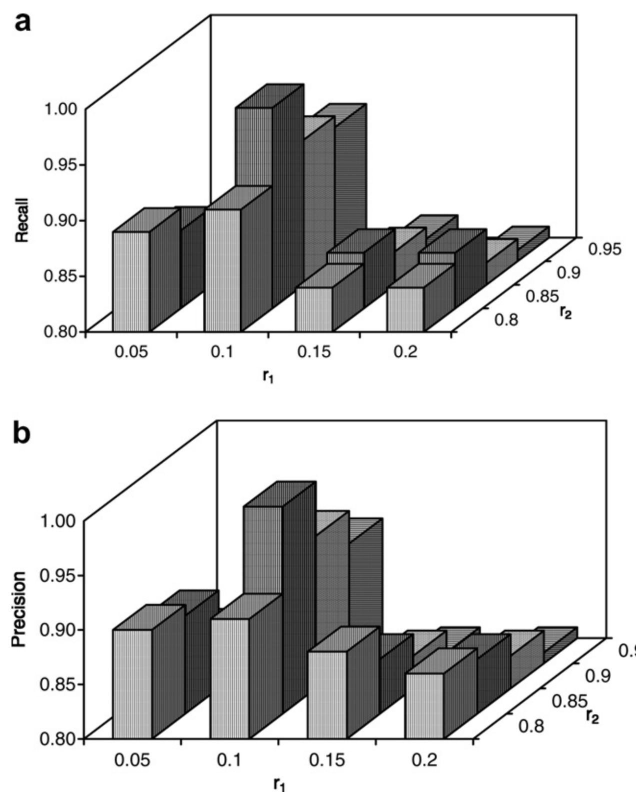

Fig. 4. ICICM based retrieval for different values of r_1 and r_2 (a) mean recall and (b) mean precision.

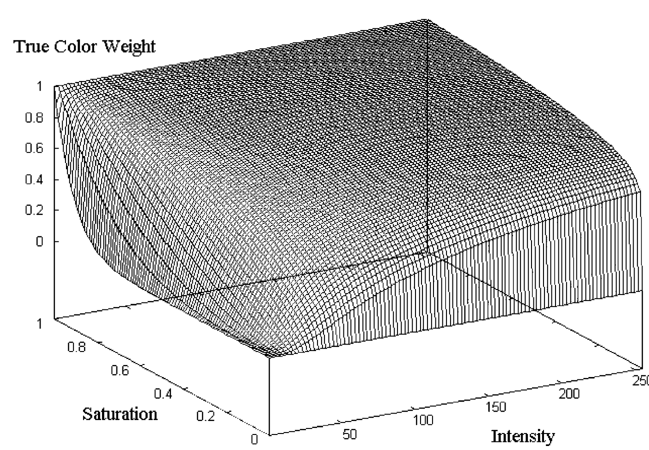

Fig. 5. Variation of $W_{col}(S, I)$ with saturation and intensity.

(0.05, 0.8)–(0.15, 0.9) give good retrieval results. For ease of understanding we use the values of HL and GL instead

Table 2
Precision vs. recall for different combinations of (r_1, r_2) with HL = 2 and GL = 6

r_1	r_2	Recall									
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
		Average precision									
0.05	0.8	0.96	0.94	0.91	0.88	0.85	0.84	0.80	0.78	0.71	0.68
0.05	0.85	0.97	0.95	0.92	0.90	0.89	0.88	0.85	0.82	0.81	0.80
0.05	0.9	0.96	0.94	0.90	0.87	0.84	0.82	0.78	0.74	0.70	0.67
0.1	0.8	0.99	0.96	0.94	0.91	0.88	0.85	0.84	0.80	0.78	0.71
0.1	0.85	**0.99**	**0.97**	**0.95**	**0.92**	**0.9**	**0.89**	**0.88**	**0.85**	**0.82**	**0.81**
0.1	0.9	0.99	0.96	0.94	0.90	0.87	0.84	0.82	0.78	0.74	0.70
0.15	0.8	0.96	0.94	0.91	0.88	0.85	0.84	0.80	0.78	0.71	0.68
0.15	0.85	0.97	0.95	0.92	0.9	0.89	0.88	0.85	0.82	0.81	0.80
0.15	0.9	0.96	0.94	0.90	0.87	0.84	0.82	0.78	0.74	0.70	0.67

Table 3
Precision vs. recall for different combinations of (r_1, r_2) with HL = 4 and GL = 4

r_1	r_2	Recall									
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
		Average precision									
0.05	0.8	0.91	0.88	0.86	0.84	0.80	0.76	0.73	0.69	0.67	0.66
0.05	0.85	0.90	0.88	0.85	0.84	0.78	0.75	0.73	0.68	0.65	0.64
0.05	0.9	0.91	0.88	0.86	0.84	0.80	0.76	0.73	0.69	0.64	0.63
0.1	0.8	0.96	0.94	0.92	0.89	0.86	0.85	0.81	0.76	0.74	0.70
0.1	0.85	**0.96**	**0.96**	**0.92**	**0.90**	**0.88**	**0.85**	**0.82**	**0.79**	**0.77**	**0.74**
0.1	0.9	0.96	0.94	0.91	0.88	0.86	0.84	0.80	0.76	0.73	0.69
0.15	0.8	0.94	0.91	0.88	0.86	0.84	0.80	0.76	0.73	0.69	0.67
0.15	0.85	0.93	0.90	0.88	0.85	0.84	0.78	0.75	0.73	0.68	0.65
0.15	0.9	0.93	0.91	0.88	0.86	0.84	0.80	0.76	0.73	0.69	0.64

of the corresponding values of Q_H and Q_I. We have further cross-validated the results by studying the retrieval performance for different combinations of r_1, r_2, Q_H and Q_I on a subset of the texture database. The results on this validation dataset are shown in Tables 2 and 3. It is noted that for $r_1 = 0.1$ and $r_2 = 0.85$, we get better average precision for both the combinations of HL and GL compared to other values of r_1 and r_2. Also, for each combination of r_1 and r_2, the (HL, GL) combination of (2, 6) gives better result.

3.3. Web-based image retrieval application

We have developed a web-based application for content-based image retrieval using ICICM. The application is available in the public domain[2]. A query in our application is specified by an example image. The number of images to be retrieved and displayed can be selected as an input parameter from the page. Various distance measures can also be chosen for ranking the images. Since image features typically have a large number of dimensions, several properties of high-dimensional Euclidean space affect the usefulness of various distance metrics. It has been shown that in CBIR applications, Euclidean distance and Vector Cosine Angle distance have similar performance in nearest neighbor queries (Qian et al., 2004). This is due to the fact that for a given query, ranking of a set of vectors by these two distance metrics is similar if the variance in the vector norms is low. In high dimensions, variance of norms of feature vectors tends to be smaller. In an earlier work, we have compared the retrieval performance of various distance measures, namely, Manhattan distance, Euclidean distance, Vector Cosine Angle distance and Histogram Intersection distance on a large image database (Vadivel et al., 2003). It was observed that although the results are comparable, Manhattan distance achieved a higher precision of retrieval. The number of color and intensity levels that can be used for generating ICICM of query image can also be dynamically selected for each query from the interface. For a given query image, the nearest neighbor result set is retrieved from the database and displayed. The value displayed below the retrieved images is the actual vector-to-vector distance computed by using the measure selected from the option. Depending on the distance metric chosen, the value is either a similarity measure (for Histogram Intersection and Vector Cosine Angle Distance) or a dissimilarity measure (for Euclidean and Manhattan Distance). The retrieval process considers all the parameters selected in the options boxes of the web page. Our applica-

[2] http://www.imagedb.iitkgp.ernet.in/icicm.php

tion provides a utility to upload an external image file and use the image as a query on our database. Thus, our application is different from all other reported research work in the field of content-based image retrieval since we provide a platform for repeating our experiments as well as for running new queries with images provided by the reader.

4. Retrieval performance

We have used two separate of image databases for measuring the performance of ICICM in content-based image retrieval applications. The first is a database contains 10,000 images from IMSI[3] master clips. The second database was generated by crawling 28,000 images from the World Wide Web. In the next two sub-sections, we present our experimental results using these databases.

4.1. Retrieval performance with labeled database

In the first set of experiments, we use general-purpose images obtained from IMSI. The images are available on CDs with some pre-classification. Similar types of images are kept in the same folder structure. These images were manually checked, re-classified if necessary, and finally a database of 10,000 images was created contains 71 classes. Each class contains between 50 and 600 images. We also selected a set of 2500 query images out of these 10,000 images for computing recall and precision. For each query image, the ground-truth has been generated by manually determining the relevant set in advance. Ten independent observers crosschecked these relevant sets and errors in categorization were corrected based on feedback from the observers before performing the experiments.

We compare the retrieval results of the proposed method with two other Co-occurrence matrix-based approaches, namely, MCCM and MCM. We also consider the Color Structure Descriptor (CS) based method-an MPEG-7 color structure descriptor proposed by Messing et al. (2001), two color histograms, namely, H(RG) and Hue (G) proposed by Gevers and Stokman (2004) as well as the LFT-based method described in Section 1. In Fig. 6, mean precision of retrieval for different values of recall have been plotted where the mean value was computed from the set of 2500 query images with ground-truth. From this figure it is noted that the performance of CS is close to MCM. It is also observed that ICICM outperforms LFT and MCCM by more than 15%. MCM also has high precision of retrieval. But it is still about 4–5% less compared to ICICM. Another interesting observation is that, performance of MCCM falls sharply with this set of images.

Besides having high value of mean precision, it is also important for image retrieval systems to have low standard deviation of precision so that the quality of retrieval

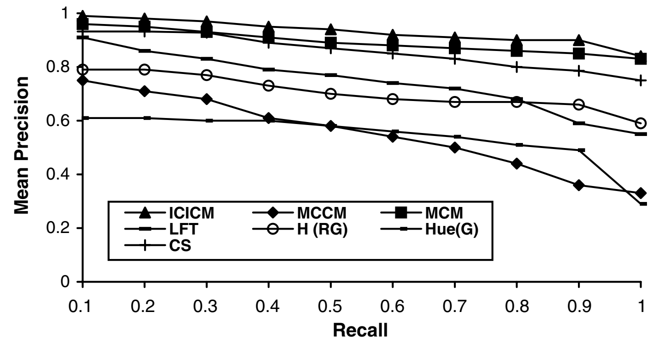

Fig. 6. Recall vs. mean precision on a labeled database of 10,000 images.

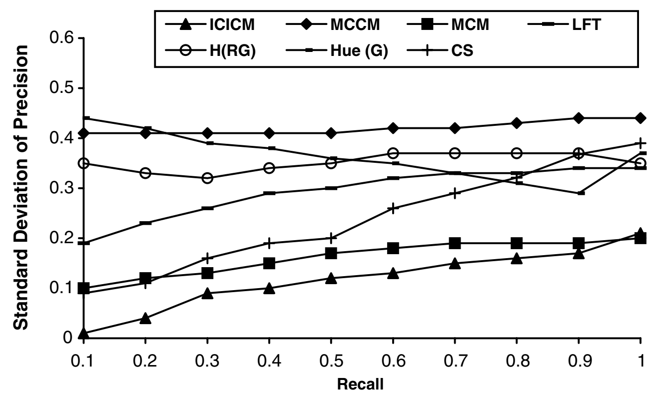

Fig. 7. Recall vs. standard deviation of precision on a labeled database of 10,000 images.

appears equally good to different observers using different query images. To study this, standard deviation of precision for all the four methods has been measured and shown in Fig. 7. It is observed that standard deviation of precision for ICICM is low compared to that of the other methods. MCM and CS also have a low value of standard deviation. However, LFT and MCCM do not exhibit good performance for this set of results.

4.2. Retrieval performance with unlabelled database

We have also tested the performance of the ICICM based image retrieval system with an even larger database of about 28,000 images. These images have been obtained using a crawler from various Internet sites, having no priori classification. Since no ground truth on the relevant sets is available on such a large database, it is not possible to determine the recall and precision values. However, for estimating retrieval performance on this unlabelled database, we considered visual content of retrieved result for measuring precision. Given a query, the number of visually close images for 2, 5, 10, and 20 nearest neighbors were used for determining precision of retrieval. In order to reduce subjectivity, we chose 50 images as queries with different visual content for performing experiments. Ten independent users were given five images each as query. Each user, using their five images, determined the precision

[3] International Microcomputer Software Inc. (http://www.imsisoft.com/)

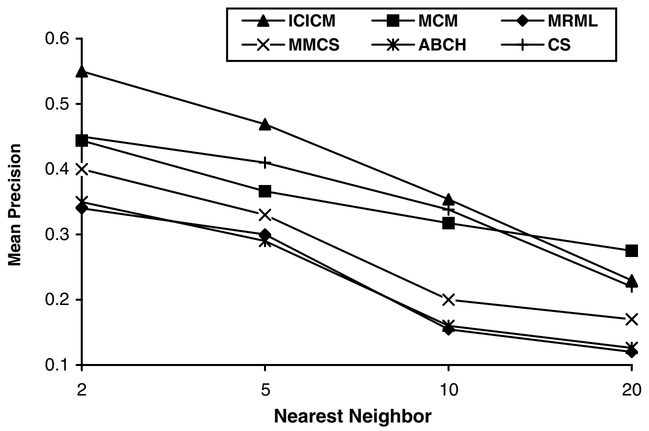

Fig. 8. Mean precision of retrieval for a large database of 28,000 images.

based on visual content for various nearest neighbors for each type of feature. Finally, the average precision was computed.

The average precision values for 2, 5, 10 and 20 nearest neighbors are shown in Fig. 8. Since MCM and CS were found to have good performance in the previous set of results, we consider them in this set also. A few other alternative methods have also been added in this comparison which include Multimedia Retrieval Markup Language with four-level relevance feedback (MRML) proposed by Mueller et al. (2001), the Multiple Multi-channel CBIR System (MMCS) proposed by French et al. (2003), the Adaptive Binning Color Histogram (ABCH) technique proposed by Leow and Li (2004). We also consider a method using a combination of color histogram and Daubechies' wavelet based texture feature (Vadivel et al., 2004b). It is noted that, when tested with a large image database having wide variety of images, precision of the image retrieval techniques deteriorates considerably. This is in contrast to our results shown in Fig. 6 on 10,000 images, which have been categorized into different groups. This can be explained by noting that in the labeled database of 10,000 images, a number of relevant images for each query are already present. A large percentage of these relevant images are retrieved, resulting in high precision and recall values. However, even with an uncontrolled image database, performance of ICICM is found to be better compared to the other approaches.

5. Conclusions

We have proposed an integrated color and intensity based co-occurrence matrix and shown its usefulness in image retrieval applications. We effectively use some of the properties of the HSV space and their relationship to human visual perception for representation of color and intensity in the co-occurrence matrix. Each pixel and its neighbor contribute to both color as well as gray level perception in the neighborhood. Four components of the co-occurrence matrix are updated with relative weights of color and gray level perception. The relative numbers of color and intensity levels can be tuned depending on the nature of the image database.

We have made extensive comparative studies of the proposed method with other recently proposed variants of the co-occurrence matrix that consider color and intensity. A few techniques that do not use co-occurrence matrix but have reported good performance, were also considered for comparison. A web-based application has been developed which is freely available to the readers for performing experiments with images in our database as well as with externally uploaded images.

The proposed color-texture descriptors can be suitably used for other image processing and pattern recognition problems as well. We would like to apply the proposed approach of generating ICICM in a number of image classification applications in future.

Acknowledgements

The work done by Shamik Sural is supported by research grants from the Department of Science and Technology, India, under Grant No. SR/FTP/ETA-20/2003 and by a grant from IIT Kharagpur under ISIRD scheme No. IIT/SRIC/ISIRD/2002–2003. Work done by A.K. Majumdar is supported by a research grant from the Department of Science and Technology, India, under Grant No. SR/S3/EECE/024/2003-SERC-Engg.

References

Aksoy, S., Haralick, R.M., 2000. A Weight Distance Approach to Relevance Feedback. In: Proc. Internat. Conf. on Pattern Recognition, pp. 812–815.

Deng, Y., Manjunath, B.S., 2001. Unsupervised segmentation of color-texture regions in images and video. IEEE Trans. Pattern Anal. Machine Intell., 800–810.

French, J.C., Watson, J.V.S., Jin, X., Martin, W.N., 2003. Integrating multiple multi-channel CBIR systems. In: Proc. Internat. Workshop on Multimedia Information Systems (MIS 2003), pp. 85–95.

Gevers, T., Stokman, H.M.G., 2004. Robust histogram construction from color invariants for object recognition. IEEE Trans. Pattern Anal. Machine Intell. 26, 113–118.

Gonzalez, R.C., Woods, R.E., 2002. Digital Image Processing. second ed.. Pearson Education.

Haralick, R.M., Shanmugam, K., Dinstein, I., 1973. Textural features for image classification. IEEE Trans. Systems Man Cybernat. 3 (6), 610–621.

Huang, J., Ravi Kumar, S., Mandar, M., Wei-Jing, Z., Ramin, Z., 1997. Image indexing using color corellograms. In: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 762–768.

Leow, W.K., Li, R., 2004. The analysis and applications of adaptive-binning color histograms. In: Proc. Computer Vision and Image Understanding (CVIU), 94(1–3), 67–91.

Manjunath, B.S., Jens-Rainer, O., Vasudevan, V.V., Yamada, A., 2001. Color and texture descriptors. IEEE Trans. Circuits Systems Video Technol. 11 (6), 703–715.

Messing, D.S.,Van Beek, P., Errico, J.H., 2001. The MPEG-7 colour structure descriptor: Image description using color and local spatial information. In: Proc. Internat. Conf. on Image Processing, vol. 1, pp. 670–673.

Mueller, H., Mueller, W., Marchand-Maillet, S., Squire, D., Pun,T., 2001. A Web-Based Evaluation System for CBIR. In: Proc. Third Internat. Workshop on Multimedia Information Retrieval, pp. 50–54.

Palm, C., 2004. Color texture classification by integrative co-occurrence matrices. Pattern Recognition 37 (5), 965–976.

Qian, G., Sural, S., Gu, Y., Pramanik, S., 2004. Similarity between euclidean and cosine angle distance for nearest neighbor queries. In: Proc. ACM Symposium on Applied Computing, pp. 1232–1237.

Shim, S., Choi, T., 2003. Image indexing by modified color co-occurrence matrix. In: Proc. Internat. Conf. on Image Processing, pp. 14–17.

Swain, M., Ballard, D., 1991. Color indexing. Internat. J. Comput. Vision 7, 11–32.

Vadivel, A., Majumdar, A.K., Sural, S., 2003. Performance comparison of distance metrics in content-based image retrieval applications. In: Proc. of Internat. Conf. on Information Technology, Bhubaneswar, India, pp. 159–164.

Vadivel, A., Sural, S., Majumdar, A.K., 2004a. Color-texture feature extraction using soft decision from the HSV color space. In: Proc. Internat. Symposium on Intelligent Multimedia Processing (ISIMP), Hong Kong, October, pp. 161–164.

Vadivel, A, Majumdar, A.K, Sural, S., 2004b. Characteristics of weighted feature vector in content-based image retrieval applications. In: Proc. IEEE Internat. Conf. on Intelligent Sensing and Information Processing, Chennai, India, pp. 127–132.

Vadivel, A., Sural, S., Majumdar, A.K., 2005. Human color perception in the HSV space and its application in histogram generation for image retrieval. In: Proc. SPIE, Color Imaging X: Processing, Hardcopy, and Applications, vol. 5667, pp. 598–609.

Wu, Y., Zhang, A., 2002. A feature re-weighting approach for relevance feedback in image retrieval. In: Proc. IEEE Internat. Conf. on Image Processing, pp. 581–584.

Yu, H., Li, M., Zhang, H.J., Feng, J., 2002. Color texture moments for content-based image retrieval. In: Proc. Internat. Conf. on Image Processing, pp. 929–931.