International Scholarly Research Network
ISRN Computer Graphics
Volume 2012, Article ID 248285, 11 pages
doi:10.5402/2012/248285

Research Article

Image Retrieval Based on Content Using Color Feature

Ahmed J. Afifi and Wesam M. Ashour

Computer Engineering Department, Islamic University of Gaza, 108 Gaza, Palestine

Correspondence should be addressed to Ahmed J. Afifi, ajafifi@iugaza.edu.ps

Received 9 December 2011; Accepted 2 January 2012

Academic Editors: T. Calvert and S. Yokoi

Content-based image retrieval from large resources has become an area of wide interest in many applications. In this paper we present a CBIR system that uses Ranklet Transform and the color feature as a visual feature to represent the images. Ranklet Transform is proposed as a preprocessing step to make the image invariant to rotation and any image enhancement operations. To speed up the retrieval time, images are clustered according to their features using *k*-means clustering algorithm.

1. Introduction

Imaging has played an important role in our life. Our ancestors used the walls of their caves to paint some pictures telling us some information about their life. With the beginning of the twentieth century, imaging has grown with an unparalleled way in all our walks of life. Now, images play an important role in many fields such as medicine, journalism, education, and entertainment. By using computers, a wide range of techniques helps us for image capturing, processing, storage, and transmitting. The emergence of the World Wide Web enables users to access data from any place and provides the exploitation of digital images in any fields [1]. Naturally, when the amount of data becomes larger and larger, it will be useless unless there are effective methods to access. Problems with effective search and navigation through the data can be solved by information retrieval.

From the past and until now, images from different fields are stored in groups as image databases. For each field, thousands of images have some common features. If we want to search for a certain image, the primitive methods, like search by text or image description, are not accurate and time consuming. These images are used in all fields such as medical, engineering, educational, sports, criminal, and many of them. For example, images in medical fields such as X-ray images are used for diagnoses and research purpose. For criminal field, face recognition is used for retrieving the suspicious people. As we mentioned before, to search for an image in a huge image database, it is not efficient to use text or image descriptors. To overcome this problem, a new technique called content-based image retrieval is used to search and retrieve an image from the database [2].

Content based image retrieval (CBIR) is the retrieval of images based on their visual features such as color, texture, and shape [3]. The first use of the concept content-based image retrieval was by Kato to describe his experiments for retrieving images from a database using color and shape features. CBIR systems have become a reliable tool for many image database applications. There are several advantages of CBIR techniques over other simple image retrieval approaches such as text-based retrieval techniques. CBIR provided solution for many types of image information management systems such as medical imagery, criminology, and satellite imagery.

A typical CBIR uses the contents of an image to represent and access. CBIR systems extract features (color, texture, and shape) from images in the database based on the value of the image pixels. These features are smaller than the image size and stored in a database called feature database. Thus the feature database contains an abstraction (compact form) of the images in the image database; each image is represented by a compact representation of its contents (color, texture, shape, and spatial information) in the form of a fixed length real-valued multicomponent feature vectors or signature. This is called offline feature extraction [4]. The main advantage of using CBIR system is that the system uses image features instead of using the image itself. So, CBIR is cheap, fast, and efficient over image search methods.

Research in CBIR is a hot topic nowadays. It is important to realize that image retrieval does not entail solving the general image understanding problems. The retrieval system presents similar images. The user should define what the similarity between images has to be. For example, segmentation and complete feature descriptors may not be necessary for image similarity. So, when we want to develop an efficient algorithm for CBIR, some problems have to be solved. The first problem is to select the image features that will represent the image. Naturally, images are endowed with information and features that can be used for image retrieval. Some of the features can be visual features (color, texture, shape), and some can be the human description of the image like impressions and emotions. The second problem is the computational steps to extract features of an image and to deal with large image databases. We have to keep in our mind that large image databases are used for testing and retrieving.

The rest of the paper is organized as the following. Section 2 reviews some related work and CBIR systems. Section 3 presents Ranklet Transform and its main role as a preprocessing step. In Section 4, we introduce the proposed system for CBIR. System implementation and experimental results are given in Section 5. Section 6 summarizes our proposed system and some proposed future work.

2. Review of Related Works

CBIR describes the process of finding images from a large data collection that match to a given query image. One highly problem when designing CBIR system is to make a system general purpose. This problem appears because of the size of the database, the difficulty of understanding images by users and computers, the difficulty of system evaluation, and results retrieval. Many general-purpose systems have been developed. QBIC [5], VIR [6], and AMORE [7] are examples of commercial general-purpose systems. Recently, academic systems have been developed such as MIT Photobook [8]. Berkeley Blobworld [9], Columbia Visualseek and Webseek [10], Netra [11], and Stanford WBIIS [12] are some of the recent well-known systems.

When a user intends to access a large image database, a linear browsing is not practical for finding the target image. Depending on the query format, image retrieval algorithms are divided into two categories: keyword-based approaches and content-based methods. In keyword-based approach, images are indexed using a keyword stored for the image describing the image content. Keyword-based retrieval is not standardized because different users describe images using different keywords. Using this approach, humans are required to personally describe every image in the database, so for a large image database the technique is cumbersome, expensive, and labor intensive. In content-based methods, the content of the image is used for search and retrieve images. This method was introduced to overcome the problem of keyword-based approach and support effective searching and browsing of large digital image libraries based on automatically derived image features.

All CBIR systems view the query image and the target images as a collection of features. These features, or image signatures, characterize the content of the image. The advantages of using image features instead of the original image pixels appear in image representation and comparison for retrieving. When we use the image features for matching, we almost do compression for the image and use the most important content of the image. This also bridges the gaps between the semantic meaning of the image and the pixel representation.

In general, features fall into two categories. The first category is global features. Global features include color and texture histograms and color layout of the whole image. The second category is local features. Local features include color, texture, and shape features for subimages, segmented regions, and interest points. These features extracted from images are then used for image matching and retrieving. The formulation of the similarity measure varies greatly. The similarity measure quantifies the resemblance in contents between a pair of images.

One of the most straightforward visual features of an image is the color because human eye is sensitive to colors. Color features are the basic characteristic of the content of images. Using color features, human can recognize most images and objects included in the image. Images added to the database have to analyze first. Images can be represented by color histograms that show the proportion of pixels of each color within the image. The most common form of the histogram is obtained by splitting the range of the data into equally sized bins. The numbers of pixels that have the same color value is computed for each bin. Therefore, it is common to use color features for image retrieval.

Several methods for retrieving images on the basis of color similarity have been proposed, but most are variations on the same basic idea. Each image added to the database is analyzed to compute its feature. Two traditional approaches have been used. Global color histogram (GCH) is used for representing images by their histograms, and the similarity between two images will be determined by the distance between their color histogram. This approach does not represent the image adequately. Furthermore, this approach is sensitive to intensity variations, color distortions, and cropping. Local color histograms (LCH) divide images into blocks and obtain the histogram for each block individually. So, an image will be represented by these histograms. To compare between two images, each block from one image will be compared with another block from the second image in the same location. The distance between these two images will be the sum of all distances. This approach represents the image more deeply and enables the comparison between image regions.

In 1996, Stricker and Dimai [13] proposed CBIR system that partitioned the image into five overlapped blocks and computed the color moments for each block. The computation was done using the HSV color space for each channel. These weights are to make the effect of one moment in a certain color channel less or greater than other moments in the same color channel. The reason to include weights in the formula is to make the effect of pixels close to the boarder less than the pixels close to the center of a certain block. In 2000, Stehling et al. [14] developed a method based on color histograms. They used a variable number of histograms called color-shape histograms (CSHs) instead of using a fixed number of cells or a single histogram. This variable histogram depends on the actual number of image colors. This method is used to represent the spatial distribution of the color for each image cell, and nonexisting color will not be presented. In 2002, Shih and Chen [15] proposed the partition-based color-spatial method. The query image is divided into 100 blocks, and for each block the three color moments are computed for each color layer. The mean vector of each block is considered as the primitive of the image. This method is not suitable for image databases containing large images. In 2004, Zhang [16] computed color and texture features from the image database. So, for each image, color and textures features are computed. When a query image is submitted, the color and texture features are computed. Firstly, the images in the database are ranked according to the color features. The top-ranked images from color features are reranked according to the texture features. In 2006, Mahdy et al. [17] proposed to extract the features from the image using its color histogram. The image is segmented into four blocks and converted from RGB color space to CIE XYZ color space then to LUV color space. The histogram is then calculated from the last color space for each of the four blocks. A histogram similarity measure is used to compare images. In 2009, Lin et al. [18] proposed a smart method for image retrieval.

Three image features and a feature selection technique are used for that. The first and the second image features are for color and texture features that are, respectively, called color cooccurrence matrix (CCM) and difference between pixels of scan pattern (DBPSP). The third image feature is for color distribution called color histogram for k-mean (CHKM). The feature selection technique is used to select the most optimal features for maximizing the detection rate and simplifying the computation of image retrieval. The three image features are not affected by image displacement and rotation and are also able to resist noise-induced variations. However, Chan and Chen [19] proposed a similar method of considering the mean value of the color component at each block. They divided the image into 3 × 3 blocks instead of 100 blocks. The mean value is computed for each block color layer (R, G, and B) separately. Although this method reduces the effect of noise and variations in size, it is affected by the shift of objects in the image. In 2008, another system is proposed by Mustaffa et al. based on color-spatial features [20]. Images in the database used for evaluation are set to the same size (192 × 128), image format (JPEG), and color space (RGB). These constraints are needed for the application and during the retrieval time. Through feature extraction, the color features are extracted using the dominant color region segmentation that maps the color in the image with the 25 color categories that can be found in the image. The region's location feature is extracted using the improved Subblock method. The extracted features are used to create the index table for all possible for that image. Some proposed systems combined color and texture features to improve the system performance. In 2010, Kekre and Mishra [21] presented a new algorithm for digital search and retrieval. They used Fast Fourier Transform for each image color components (R, G, and B). The frequency plane for each component is divided into 12 sectors. The real and the imaginary parts for each sector are computed, and their average is taken as one parameter for the feature vector that will represent the image. In March 2011, Sharma et al. [22] proposed an efficient CBIR using color histogram processing. The proposed method used color moments and color histograms. They computed the color moments from the images. The color features are used for matching between the query image and images in the database. They also computed the color histograms from the color components of the query image and compare them with a number of top-ranked images from the first step.

As a summary, dealing with color images and extracting features have some drawbacks. First, color images have large dimensions, and the computations are quite time consuming. Second, color images are sensitive to noise interference as illumination. Furthermore, most CBIR systems cannot handle rotation and translation. Our contribution is to overcome most of the previous problems. We propose a color-based retrieval system for comparing similarities between images. Our proposed system will reduce the computation of the distance to find the similarity between images. We will address the issue of indexing for the image database we will use. The system will be able to overcome the translation and rotation.

3. Ranklet Transform

Fourier Transform has been the mainstay of signal transform. Fourier Transform converts a signal from the time domain into the frequency domain to measure the frequency components of the signal. For CBIR, Fourier Transform was used to extract texture features from high-frequency components of the image. Unfortunately, Fourier Transform failed to capture the information about objects locations in an image and could not provide local image features. After its revolution, Wavelet Transform was used for image processing methods due to its efficiency and effectiveness in dealing with most important image processing tasks such as image analysis and compression. The main difference between Fourier Transform and Wavelet Transform is how to represent the signals to be analyzed. Fourier Transform represents the signal as weighted sum of the sinusoidal trigonometric functions. On the other hand, Wavelet Transform uses completely supported functions of limited duration in time and frequency domains [23].

Ranklet Transform belongs to a family of nonparametric, orientation-selective, and multiresolution features that has the wavelet style. It has been used for pattern recognition and in particular to face detection. Later on, it has been used for testing and estimating 3D structure and motion of objects. From 2004, Ranklet Transform has been used in medical fields. It has been applied to the problems of tumoral masses detection in digital mammograms. Some tests show that Ranklet Transform performs better than some methods such as pixel-based and wavelet-based image representations. Ranklet Transform has three main properties. First, it is nonparametric that it is based on nonparametric statistics that deal with a relative order of pixels instead of their intensity values. Second, it is orientation selective that it is modeled on Haar wavelets. This means that for an image, vertical, horizontal, and diagonal ranklet coefficients are computed. Finally, it is multiresolution that the Ranklet Transform can be calculated at different resolutions using Haar wavelet supports. Now, Ranklet Transform properties are discussed in detail [24].

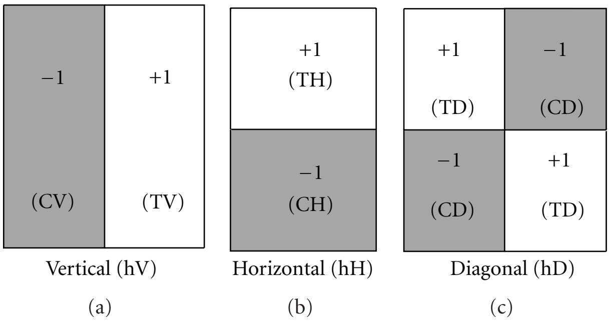


Figure 1: Ranklet Transform orientation-selective analysis.

3.1. Nonparametric Statistics. The expression non-parametric denotes statistical methods that are distribution-free from the data of a given probability distribution. Non-parametric methods are useful when the applications are interested in ranking rather than numerical interpretation. The robustness of non-parametric methods makes scientists apply them in several learning methods such as support vector machine (SVM) [25].

Suppose a set S of $P_1, P_2, \ldots, P_N$ pixels and we want to perform a rank transform [26] on the set. We refer to the rank transform with the symbol π. The rank transform will order the elements in the set in an ascending order and substitute each pixel value with its rank among all other pixels. For example, if we have the matrix S of size 4×4,

$$S = \begin{bmatrix} 55 & 99 & 25 & 153 \\ 26 & 75 & 92 & 200 \\ 21 & 64 & 88 & 154 \\ 101 & 190 & 199 & 222 \end{bmatrix}. \tag{1}$$

If we apply the rank transform on S we get:

$$\pi(S) = \begin{bmatrix} 4 & 9 & 2 & 11 \\ 3 & 6 & 8 & 15 \\ 1 & 5 & 7 & 12 \\ 10 & 13 & 14 & 16 \end{bmatrix}. \tag{2}$$

The Rank Transform and Wilcoxon Test [25] are related. Consider a set of N pixels that divides into two sub-sets. The first sub-set is called treatment (T) that contains m pixels, and the second subset is called control (C) that contains n pixels, so that $m+n = N$. We have to test whether the T set is significantly higher or lower than the C set. We first rank the elements in each subset (T and C). Let W_S be the Wilcoxon statistic defined by

$$W_S = \sum_{i=0}^{N} \pi_i V_i, \tag{3}$$

where π_i Rank of element i

$$V_i = \begin{cases} 0, & \pi_i \in \mathrm{C}, \\ 1, & \pi_i \in \mathrm{T}. \end{cases} \tag{4}$$

From (4), we are just interested in the sum of ranks for the set T. We say that a set T is higher than a set C if W_S is greater than critical value τ, in other words if $W_S > \tau$. In order to deal with something equivalent to Wilcoxon test but with an immediate interpretation in terms of pixels comparison, Mann Whitney test (W_{XY}) is defined as

$$W_{XY} = W_S - \frac{m(m+1)}{2}. \tag{5}$$

Equation (5) is equal to the number of pixel pairs (p_m, p_n) with $p_m \epsilon$ T and $p_n \epsilon$ C, such that the intensity value of p_m is higher than the intensity value of p_n. The value of W_{XY} ranges from 0 to the number of pairs $(p_m, p_n) \epsilon \mathrm{T} \times \mathrm{C}$, namely, $m \times n$. Notice that to compute the value of W_{XY}, it takes a huge computational time approximately $O(N^2)$. To reduce this computational time, the value is obtained by the rank transform that ranks the pixels and sum the ranks of the treatment set T.

3.2. Orientation Selectivity. The second property of Ranklet Transform is orientation selective. This property is derived from the fact that it is modeled based on bidimensional Haar wavelets [27]. Haar wavelets supports divide a 2D set into two equally subsets. They divide the set in different orientation. Suppose an image containing N pixels. To compute the Mann Whitney test, the N pixels is divided into two subsets T and C of size $m = n = N/2$. Thus, half of pixels are assigned to the set T, and the second half of pixels are assigned to the set C. As we said, we can divide the set into different orientations. We are basically interested in dividing the set in vertical, horizontal, and diagonal orientation. This is similar to the Haar wavelet supports that divide the set into vertical Haar (h_V), horizontal Haar (h_H), and diagonal Haar (h_D). Figure 1 [24] illustrates the three Haar wavelet supports that any two subsets can be described using them. An important note is that the arbitrariness distribution of the two subsets T and C is the basic in order to be able to choose the two subsets based on the Haar wavelet supports. In other words, the arbitrariness with which we choose the subsets is the orientation-selective property of the Ranklet Transform.

Once we have introduced the Haar wavelet supports, the definition of the Ranklet Transform is straightforward. For an image constituted by a set of N pixels, we first derive the Haar wavelet supports in the three orientations (vertical, horizontal, and diagonal).

For each orientation we divide the set into sub-sets T and C. We compute the rank transform and then the Mann

Whitney test for each orientation. The Ranklet coefficient R_j then can be computed by the following equation:

$$R_j = \frac{W_{XY}^{j}}{mn/2} - 1, \qquad (6)$$

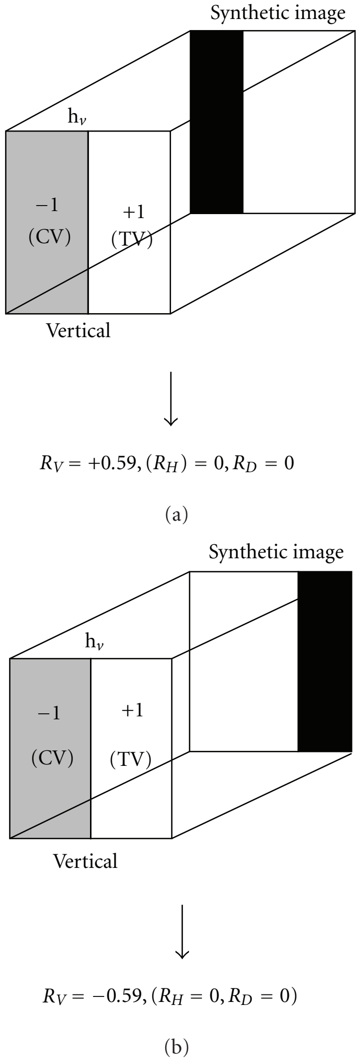
where $j = \mathrm{V},\ \mathrm{H},\ \mathrm{D}$.

Notice that W_{XY}^{j} is computed for each vertical, horizontal, and diagonal Haar wavelet supports by splitting the N pixels into sub-sets T_j and C_j. Ranklet coefficients have a geometric interpretation. Figure 2 shows a synthetic image that has a vertical edge with the darker side on the left (left part), where $\mathrm{C_V}$ is located, and the brighter side on the right, where $\mathrm{T_V}$ is located [28]. The Ranklet coefficient will be close to +1 if many pixels in $\mathrm{T_V}$ have higher intensity values than the pixels in $\mathrm{C_V}$. Conversely, the Ranklet coefficient will be close to the value −1 if the dark and bright sides are reversed. The same idea can be drawn for the other Ranklet coefficients in alternative orientation (horizontal and diagonal).

As we mentioned before, to compute the Ranklet coefficients, we have to compute the rank transform and the Mann Whitney test to get the W_{XY}. The computation of the Ranklet coefficient using the Mann Whitney test clarifies the nonparametric property of the Ranklet Transform. From the Haar wavelet supports, we can get different orientations for the image, namely, vertical, horizontal, and diagonal. Putting the image in different orientations by means of Haar wavelet supports clarifies the orientation-selective property of the Ranklet Transform.

3.3. The Multiresolution. Because the Ranklet Transform is modeled on the Haar wavelet, this leads to extend the multiresolution property to the Ranklet Transform [28]. Ranklet coefficients can be computed at different resolutions by stretching and shifting the Haar wavelet supports. For example, suppose that we have to perform the Ranklet Transform on an image of size 8 × 8. The Ranklet Transform will be performed at different resolutions as {8, 4, and 2}. Figure 3 [28] shows the image with size 8 × 8 and the Haar wavelet supports with pixels 8×8, 4×4, and 2×2 to perform the Ranklet Transform. We also consider that the vertical and horizontal shifts of the Haar wavelet supports along the horizontal and vertical dimensions are of 1 pixel. After performing the Ranklet Transform, we find that the image will be composed by 1 triplet $R_{V,H,D}$ of Ranklet coefficients derived from the Ranklet Transform at resolution 8, 25 triplets at resolution 4, and 49 triplets at resolution 2. In general, we can calculate the number of triplets (the size of the generated vertical, horizontal, and diagonal Ranklet Images) that can be generated after performing the Ranklet Transform on an image of size $I \times I$ at a resolution $S \times S$ using.

$$n\mathrm{T} = (I + 1 - S)^2. \qquad (7)$$



(a)

(b)

Figure 2: Ranklet Transform applied to some synthetic images.

4. The Proposed Algorithm

In this section, we introduce our proposed CBIR system. In our proposed system, we will extract some color features to represent the image and use these features to compare between the images.

4.1. Color Feature Extraction. The color composition of an image can be viewed as a color distribution in the sense of the probability theory. The discrete probability distribution can be viewed as a histogram. The color histogram is one of the most well-known color features used for image feature extraction. It denotes the joint probability of the intensities of the image. From the probability theory, a probability distribution can be uniquely characterized by its moments. Thus, if we interpret the color distribution of an image as a probability distribution, moments can be used to characterize the color distribution. In our work, the moments of the color distribution are the features extracted from the images, and we will use them to describe the image and for image matching and retrieval [29].

The first-order (*mean*), the second (*standard deviation*), and the third-order (*skewness*) color moments have been

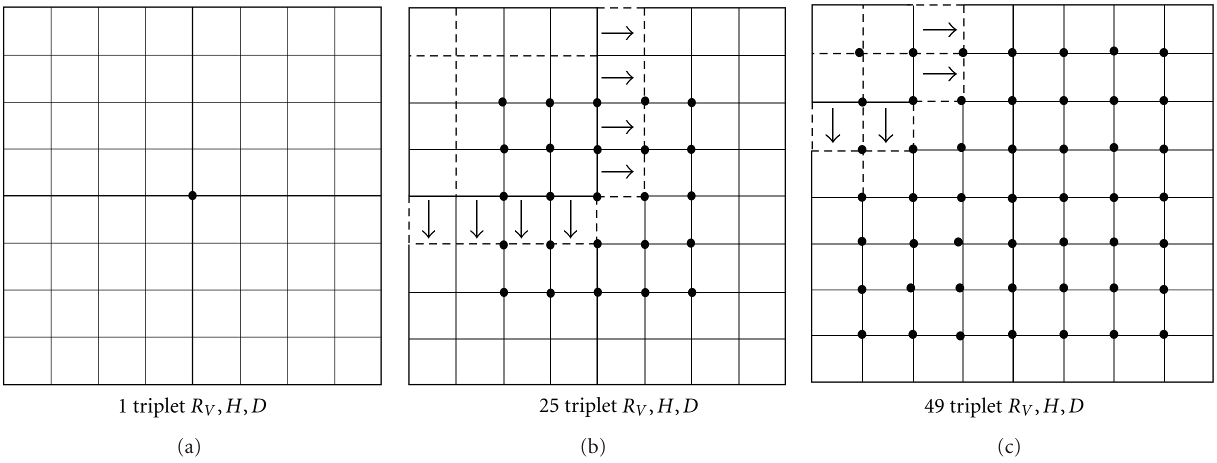


Figure 3: Multiresolution Ranklet Transform at different resolutions.

proved to be efficient and effective in representing color distributions of images. If the value of the i th color channel at the j th image pixel is p_{ij}, then the color moments are as follows.

Moment 1. Mean

$$E_i = \frac{1}{N} \sum_{j=1}^{N} p_{ij}, \tag{8}$$

Moment 2. Standard deviation

$$\sigma_i = \sqrt{\left(\frac{1}{N} \sum_{j=1}^{N} \left(p_{ij} - E_i\right)^2\right)}, \tag{9}$$

Moment 3. Skewness

$$s_i = \sqrt[3]{\left(\frac{1}{N} \sum_{j=1}^{N} \left(p_{ij} - E_i\right)^3\right)}. \tag{10}$$

For color image, color moments are very compact representation features compared with other color features since only 9 (3 values for each layer) numerical values are used to represent the color content of each image channel.

4.2. K-Means for Database Clustering. Time is one of the most important factors to CBIR. It mainly depends on the number of images in the database. Many systems use every image in the database to be compared with the query image to find the top matching images. This method is highly computationally inefficient when the database contains large number of images. However, the results will be more accurate when we use all images in the database for similarity matching. To overcome this problem, image clustering or categorization has often been used as a preprocessing step to speed up image retrieval in large databases and to improve the accuracy.

Clustering algorithms are used as a preprocessing step to cluster the database into N different categories [30].

In k-means algorithm, the clustering results are measured by the sum of within-cluster distances between every vector and its cluster centroid. This criterion ensures that the clusters generated are tight. K-means algorithm takes k, the number of clusters to be generated, as the input parameter and partitions a set of N objects into k clusters so that the resulting intracluster similarity is high but the intercluster similarity is low. If the number of clusters is not specified, a simple method is done. The algorithm initializes the number of clusters to a certain number less than the total number of the samples in the dataset. The algorithm increases that number gradually until the average distance between a vector and its cluster centroid is below a given threshold [26, 31].

The k-means algorithm works as the following. The number of clusters, k, is entered as an input parameter. The algorithm randomly selects k of the objects, each of which initially represents a cluster centroid. For each of the remaining objects, an object is assigned to the cluster to which it is most similar. Similarity between the cluster centroid and the object is determined by a distance. For example, if we use the Euclidean distance to measure the similarity and the result is smaller than a threshold, this means that this object belongs to the cluster. It then computes the new mean for each cluster. This process is iterated until the criterion function converges. The criterion function used for convergence is the sum of squared error (SSE). The SSE is defined by

$$E = \sum_{i=1}^{k} \sum_{p \in c_i} |p - m_i|^2. \tag{11}$$

In (11), E is the sum of squared error for all objects in the database, p is feature vector representing the image, and m_i is the mean of cluster c_i. This criterion attempts to make the resulting k clusters as separate as possible. The algorithm tries to determine the k clusters that minimize the squared error function. It will work well when the clusters are compact and well separated from one another. This

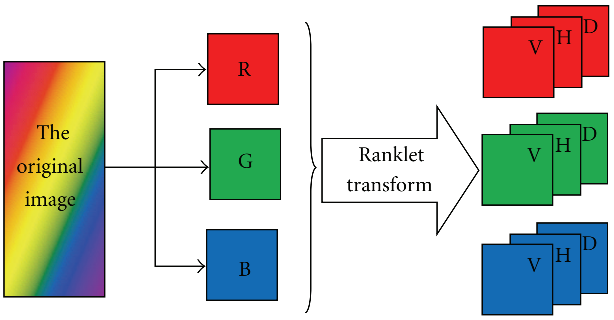


Figure 4: The preprocessing step.

algorithm is relatively scalable and efficient in processing large data sets, but it often terminates at a local optimum. The k-means algorithm is given in Algorithm 1 [31]. We use the k-means algorithm to classify the feature vectors of the input images. We select the k-means algorithm because it is suitable to cluster large amount of data. Each feature vector is treated as an object having a location in space. The cluster generates where objects within the cluster are closed to each other and far from objects in other clusters as possible. Selecting the distance measure is an important step in clustering. The distance measure determines the similarity of two images. To begin clustering, k objects are selected randomly to initialize the centroids of the clusters. The centroid for each cluster is a point to which the sum of distances from all objects in that cluster is minimized. The clusters are generated by calculating the distance between the centroid and the objects for each cluster.

4.3. The Proposed CBIR System. In this section we are introducing our proposed system for CBIR. In our system, we will use images represented in the RGB color space. Although the RGB color space is perceptually nonuniform and device dependent, it is widely used for representing color images. We do not worry about that because the image will have a preprocessing step. Also, images in the database seem to be captured by the same way (the same capturing device and environmental conditions). WANG database [32] is widely used for CBIR, and we will use it to test our system. Ranklet Transform is used as a preprocessing step. The advantage of using Ranklet Transform is that it generates three images with different orientation that are vertical, horizontal, and diagonal images. Because we use RGB color image, we perform the Ranklet Transform for each image layer (the red, green, and blue layers). The Ranklet Transform generates three images for each layer, three ranklet images for the red layer, three ranklet images for the green layer, and three ranklet images for the blue layer. Figure 4 shows the preprocessing step.

After applying the Ranklet Transform for the image, we have to extract its color features that represent the image. We have nine images generated from the preprocessing step. We calculate the three color moments for each image using (8) for the mean, (9) for the standard deviation, and (10) for the skewness. After calculating the three values, we concatenate them in one vector called image feature vector. This vector has 27 values that represent the image, and we use it to measure the similarity between the images.

We calculate the feature vector for every image in the database. We use the k-means clustering algorithm to cluster the images into different categories. This step will decrease the time of retrieval. When the user submits an image, it will not be compared with each image in the database. Instead, the feature vector of the image will be compared with the centroid of each category by computing the distance between them. The smallest distance means that the input image is similar to the category which holds that centroid. The input image then is compared with the images in that category, and system will retrieve the most similar images to the input image. The retrieved images will be ranked, and the user will specify the number of images to appear. The proposed algorithm is stated in Algorithm 2.

5. Experiments and Results

In this section, we will present the evaluation of our proposed system. As we said before, we used WANG database for system evaluation. WANG database is an image database where the images are manually selected from the Corel database. In WANG database, the images are divided into 10 classes. Each class contains 100 images. It is widely used for testing CBIR systems. Classification of the images in the database into 10 classes makes the evaluation of the system easy. Our proposed system is implemented using Matlab image processing. From the 1000 images, we randomly select 300 images from the database. From each class we randomly select 30 images. The selected images went through the implemented system to extract the features and stored them. The extracted features are clustered by the k-means clustering algorithm. This step is made off line for the 300 images selected from the database. The database now is ready for testing and evaluating our CBIR proposed system.

5.1. Performance Evaluation Metrics for CBIR Systems. When we want to evaluate a CBIR system, we may face some problems. One major problem we face for CBIR system performance evaluation is that neither a standard database nor a unique performance measure is available. There are many image databases that are used to represent results for CBIR system. So, no standard image databases are available for CBIR systems. Therefore, it is impossible to compare the performance of different systems using different image databases. Furthermore, the early CBIR systems were restricted by their results. They presented one or more example queries, which are used to give a positive impression about the efficiency of the system. This is neither a quantitative nor an objective measure that can be used for system performance evaluation.

In CBIR, the most commonly used performance measures are Precision and Recall. Precision is defined as the ratio of the number of retrieved relevant images to the total

```
Purpose: The k-means algorithm for partitioning based on the mean value of the objects in the cluster.
Input: A database of N objects, number of clusters k.
Output: A set of k clusters.
Method:
(1) Arbitrarily choose k objects as the initial cluster centers.
(2) (Re) Assign each object to the cluster to which the object is the most similar based on the mean value of objects in the cluster.
(3) Update the cluster means, that is, calculate the mean value of the objects for each cluster.
(4) Repeat steps 2 and 3 until no changes.
```

Algorithm 1: The *k*-means Algorithm.

```
Purpose: The algorithm is to retrieve images similar to the input image.
Input: An RGB image, number of retrieved images n.
Output: n images similar to the input image.
Method:
Step 1 : The input image is a color image in RGB color space.
Step 2 : Apply the Ranklet Transform for each image layer (R, G, and B). The output images will be in three orientations (vertical, horizontal, and diagonal).
Step 3 : For each ranklet image (vertical, horizontal, and diagonal) in a specified layer, calculate the color moments (8), (9), and (10).
Step 4 : Construct the feature vector that will represent the image containing 27 numerical values.
Step 5 : Cluster the images in the database using k-means algorithm (Algorithm 1) into different categories.
Step 6 : Calculate the distance between the input image and the centroid of each cluster using Euclidian Distance, and find the smallest distance.
Step 7 : Calculate the distance between the input image and the images in the cluster that has the smallest distance with the input image.
Step 8 : Retrieve the first n images that is similar to the input image.
```

Algorithm 2: The proposed CBIR algorithm.

number of retrieved images [33]. We denote the precision by *P*. The equation of precision is

$$P = \frac{\text{Number of relevant images retrieved}}{\text{Total number of images retrieved}}. \tag{12}$$

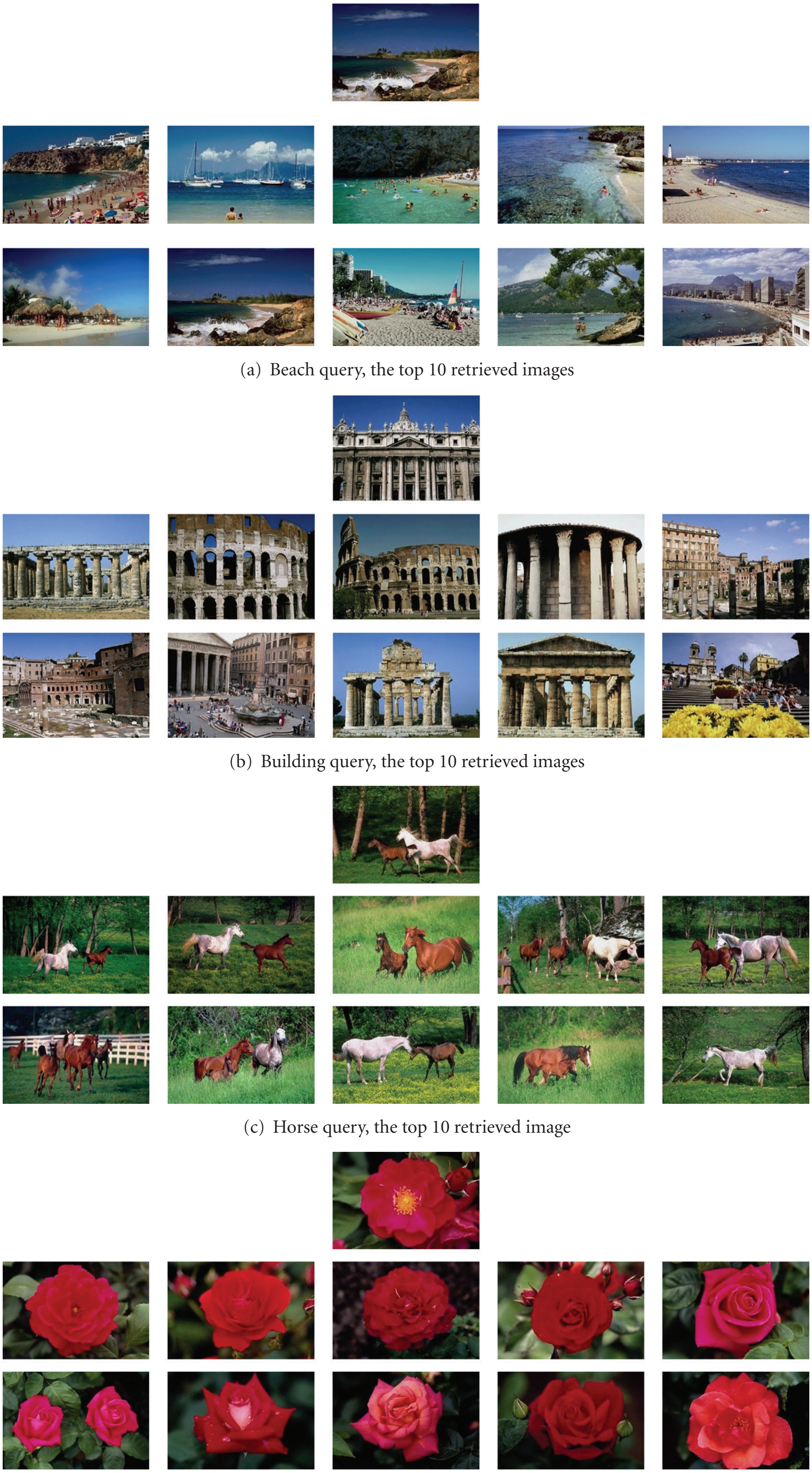
Recall is defined as the ratio of the number of retrieved relevant images to the total number of relevant images in the database [33]. We denote the recall by *R*. The equation of recall is

$$R = \frac{\text{Number of relevant images retrieved}}{\text{Total number of relevant images in the database}}. \tag{13}$$

In CBIR, if the precision score is 1.0, this means that every image retrieved by a search is relevant, but we do not know if the search retrieves all the images relevant to the query. If the recall score is 1.0, this means that all relevant images are retrieved by the search, but we do not know the number if irrelevant images were also retrieved.

5.2. Results and Evaluation. In order to check retrieval effectiveness of the proposed system, we have to test it by selecting some images randomly and retrieve some images. Also, we have to calculate the two performance evaluation metrics, precision and recall. Finally, we compare our proposed system with other existing systems and show the efficiency of the proposed system. For initial test to our proposed system, we select four images from different classes randomly and retrieve 10 images similar to the query image. Figure 5 shows the results generated from the proposed system.

One traditional graph that describes the performance of the system is the Precision-Recall graph. It provides a meaningful result when the database is known and has been used be some earlier systems. To evaluate our proposed system, we use the Precision-Recall graph. We select 30 images randomly from each class in the database to use them as queries to calculate the precision and recall. For each image, the precision of the retrieval result is obtained by increasing the number of retrieved images. Figure 6 shows the Precision-Recall graph. From the figure, we notice that the system has good precision results over the different values of recall. The maximum average precision of 0.93 at recall value is 0.05, and the precision value decreases to 0.47 at 0.28 of recall. For example, if a user submits a query image and he wants just 10 relevant images from 100 images retrieved by the system, the graph shows us that the user will get 70 relevant images to the query image instead of 10 images. In other words, for an average recall value of 10%, we have an average precision of 70%. This means that we want to get 10% of the relevant images in the database, and we will get



(a) Beach query, the top 10 retrieved images

(b) Building query, the top 10 retrieved images

(c) Horse query, the top 10 retrieved image

(d) Flower query, the top 10 retrieved images

Figure 5: Four query response examples of the proposed system.

TABLE 1: Comparison of precision of the proposed system with previously existed systems.

Semantic Group	Jhanwar et al.	Huang and Dai's	CTDCIRS	SCBIRS	Proposed System
Africans	0.4525	0.4240	0.5620	0.6830	0.7143
Beaches	0.3975	0.4455	0.5360	0.5400	0.8560
Buildings	0.3735	0.4105	0.6100	0.5615	0.8341
Buses	0.7410	0.8515	0.8930	0.8880	0.8571
Dinosaurs	0.9145	0.5865	0.9840	0.9925	0.9975
Elephants	0.3040	0.4255	0.5780	0.6580	0.7143
Flowers	0.8515	0.8975	0.8990	0.8910	0.9374
Horses	0.5680	0.5890	0.7800	0.8025	0.5714
Mountains	0.2925	0.2680	0.5120	0.5215	0.4286
Foods	0.3695	0.4265	0.6940	0.7325	0.9752

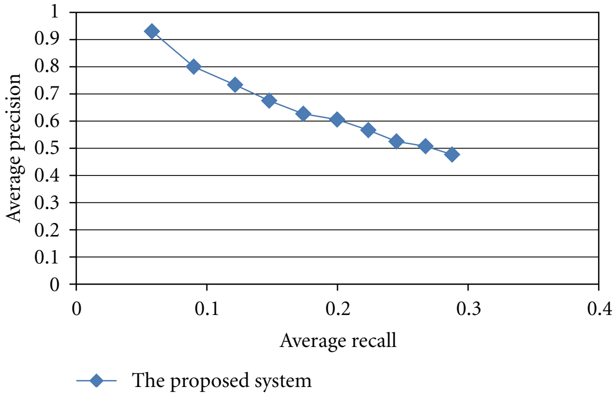


FIGURE 6: The average precision/recall chart of the proposed system.

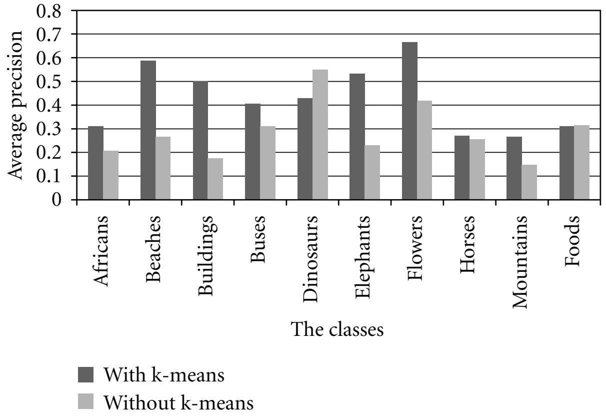


FIGURE 7: Comparison of precision of the proposed system applied with clustering and without clustering.

70% of the retrieved images that are relevant to the query image. This clarifies that our system works well.

We have improved the efficiency of our proposed system by using *k*-means clustering algorithm to cluster the database. The average precision for each class for the top 30 relevant images using *k*-means clustering algorithm is shown in Figure 7. Using *k*-means clustering algorithm, we found that the system works better than without using the clustering algorithm.

5.3. Comparison of the Proposed System with Other Systems. In this section, we present some earlier CBIR systems' results and compare them with our proposed system. The existing systems we chose for comparison use color features to represent images, and they also use the WANG database to evaluate their proposed systems. To evaluate our proposed system, we use each image in our database to be a query image and submit it to the system. We calculate the precisions for each query in all classes. Then for each class we take the average of all calculated precisions as shown in Table 1. The result of this study is compared against the performance of Jhanwar et al.'s method [34], Huang and Dai's method [35], CTDCBIRS [36], and SCBIRS [18]. Table 1 shows that our proposed system performs significantly better than other systems for all classes except for classes 4, 8, and 9 which are Buses, Horses, and Mountains, respectively.

6. Conclusion and Future Work

Nowadays, content-based image retrieval is a hot topic research. Many researches have been done to develop some algorithms that solve some problems and achieve the accuracy when retrieving images and distinguishing between them. Many proposed algorithms use images to extract features and use their features for similarity matching. However, most of the algorithms use the gray scales images. In our work, we use the color image to extract the color feature and use it for similarity matching. We do this because most images in our world are color images. So, color feature is one of the most features that can be taken into account when developing a CBIR system. In the future, we can develop a system that combines the texture, shape, and spatial features with the color feature to represent the image. This will give good results. Also, segmentation is a method to extract regions and objects from the image. The segmented regions are used for similarity matching.

References

[1] J. Eakins and M. Graham, "Content-based image retrieval," A Report to the JISC Technology Applications Programme, Institute for image database research, University of Northumbria, Newcastle, UK, 1999.

[2] H. Tamura and N. Yokoya, "Image database systems: a survey," *Pattern Recognition*, vol. 17, no. 1, pp. 29–43, 1984.

[3] B. Jerome, V. Regine, and S. Georges, *A Signature Based on Delaunay Graph and Co-Occurrence Matrix*, Laboratoire Informatique et Systematique, Universiyt of Paris, Paris, France, 2002.

[4] F. Long, H. Zhang, H. Dagan, and D. Feng, "Fundamentals of content based image retrieval," in *Multimedia Information Retrieval and Management*, D. Feng, W. Siu, and H. Zhang, Eds., Multimedia Signal Processing Book, chapter 1, pp. 1–26, Springer, Berlin, Germany, 2003.

[5] M. Flickner, H. Sawhney, W. Niblack et al., "Query by image and video content: the QBIC system," *IEEE Computer*, vol. 28, no. 9, pp. 23–32, 1995.

[6] A. Gupta and R. Jain, "Visual information retrieval," *Communications of the Association for Computing Machinery*, vol. 40, no. 5, pp. 70–79, 1997.

[7] S. Mukherjea, K. Hirata, and Y. Hara, "AMORE: a world wide web image retrieval engine," *World Wide Web*, vol. 2, no. 3, pp. 115–132, 1999.

[8] A. Pentland, R. W. Picard, and S. Sclaroff, "Photobook: content-based manipulation of image databases," *International Journal of Computer Vision*, vol. 18, no. 3, pp. 233–254, 1996.

[9] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik, "Blobworld: a system for region-based image indexing and retrieval," in *Proceedings of the Visual Information Systems*, pp. 509–516, June 1999.

[10] J. Smith and S. Chang, "Visualseek: a fully automated content-based image query system," in *Proceedings of the 4th ACM International Multimedia Conference & Exhibition*, pp. 87–98, Boston, Mass, USA, November 1996.

[11] W. Y. Ma and B. S. Manjunath, "Netra: a toolbox for navigating large image databases," in *Proceedings of the IEEE International Conference on Image Processing*, pp. 568–571, October 1997.

[12] J. Wang, G. Wiederhold, O. Firschein, and X. Sha, "Content-based image indexing and searching using daubechies' wavelets," *International Journal on Digital Libraries*, vol. 1, no. 4, pp. 311–328, 1998.

[13] M. A. Stricker and A. Dimai, "Color indexing with weak spatial constraints," in *Proceedings of the Storage and Retrieval for Still Image and Video Databases IV Conference*, vol. 2670 of *Proceedings of SPIE*, pp. 29–40, San Diego, Calif, USA, February 1996.

[14] R. Stehling, M. Nascimento, and A. Falcao, "On shapes of colors for content-based image retrieval," in *Proceedings of the ACM Workshops on Multimedia Conference*, Los Angeles, Calif, USA, November 2000.

[15] J. L. Shih and L. H. Chen, "Colour image retrieval based on primitives of colour moments," *IEE Proceedings*, vol. 149, no. 6, pp. 370–374, 2002.

[16] D. Zhang, "Improving image retrieval performance by using both color and texture features," in *Proceedings of the IEEE 3rd International Conference on Image and Graphics (ICIG '04)*, pp. 172–175, Hong Kong, December 2004.

[17] Y. Mahdy, K. Shaaban, and A. Abd El-Rahim, "Image retrieval based on content," *Journal of Graphics, Vision and Image Processing*, vol. 6, no. 1, pp. 55–60, 2006.

[18] C. H. Lin, R. T. Chen, and Y. K. Chan, "A smart content-based image retrieval system based on color and texture feature," *Image and Vision Computing*, vol. 27, no. 6, pp. 658–665, 2009.

[19] Y. K. Chan and C. Y. Chen, "Image retrieval system based on color-complexity and color-spatial features," *Journal of Systems and Software*, vol. 71, no. 1-2, pp. 65–70, 2004.

[20] M. R. Mustaffa, F. Ahmad, R. W. O. K. Rahmat, and R. Mahmod, "Content-based image retrieval based on color-spatial features," *Malaysian Journal of Computer Science*, vol. 21, no. 1, pp. 1–12, 2008.

[21] H. Kekre and D. Mishra, "CBIR using upper six FFT sectors of color images for feature vector generation," *International Journal of Engineering and Technology*, vol. 2, no. 2, pp. 49–54, 2010.

[22] N. Sharma, P. Rawa, and J. Singh, "Efficient CBIR using color histogram processing," *Signal & Image Processing*, vol. 2, no. 1, p. 94, 2011.

[23] M. Porat and Y. Y. Zeevi, "The generalized Gabor scheme of image representation in biological and machine vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 4, pp. 452–468, 1988.

[24] F. Smeraldi, "Ranklets: orientation selective non-parametric features applied to face detection," in *Proceedings of the 16th International Conference on Pattern Recognition*, vol. 16, no. 3, pp. 379–382, 2002.

[25] E. L. Lehmann, *Nonparametrics: Statistical Methods Based on Ranks*, Holden-Day, San Francisco, Calif, USA, 1995.

[26] R. Datta, J. Li, and J. Wang, "Content-based image retrieval—approaches and trends of the new age," *ACM Computing Surveys*, vol. 40, no. 2, article 5, pp. 1–60, 2008.

[27] I. Daubechies, *Ten Lectures in Wavelets*, SIAM, Philadelphia, Pa, USA, 1992.

[28] M. Masotti, "A ranklet-based image representation for mass classification in digital mammograms," *Medical Physics*, vol. 33, no. 10, pp. 3951–3961, 2006.

[29] M. Stricker and M. Orengo, "Similarity of color images," in *Storage and Retrieval for Image and Video Databases III*, vol. 2420 of *Proceedings of SPIE*, pp. 381–392, February 1995.

[30] V. Murthy, E. Vamsidhar, J. Kumar, and P. Sankara, "Content based image retrieval using hierarchical and K-means clustering techniques," *International Journal of Engineering Science and Technology*, vol. 2, no. 3, pp. 209–212, 2010.

[31] P. Jeyanthi and V. J. S. Kumar, "Image classification by K-means clustering," *Advances in Computational Sciences and Technology* , vol. 3, no. 1, pp. 1–8, 2010.

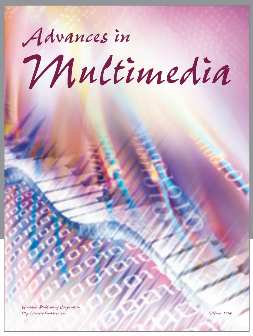
[32] J. Z. Wang, "Wang database," 2010, http://wang.ist.psu.edu/.

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

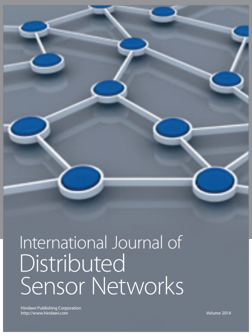
[34] N. Jhanwar, S. Chaudhuri, G. Seetharaman, and B. Zavidovique, "Content based image retrieval using motif cooccurrence matrix," *Image and Vision Computing*, vol. 22, no. 14, pp. 1211–1220, 2004.

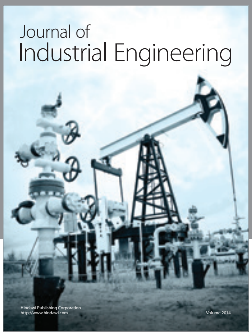
[35] P. W. Huang and S. K. Dai, "Image retrieval by texture similarity," *Pattern Recognition*, vol. 36, no. 3, pp. 665–679, 2003.

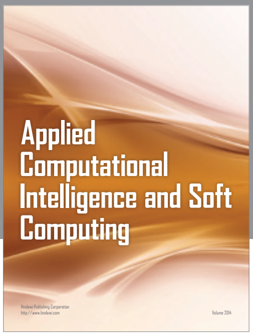
[36] M. B. Rao, B. P. Rao, and A. Govardhan, "CTDCIRS: content based image retrieval system based on dominant color and texture features," *International Journal of Computer Applications*, vol. 18, no. 6, pp. 40–46, 2011.

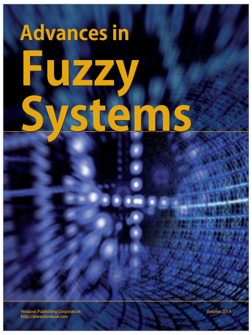

Advances in
Multimedia

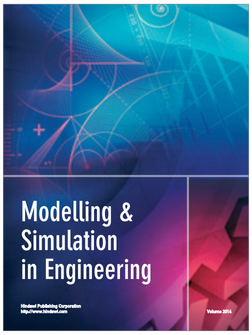
The Scientific
World Journal

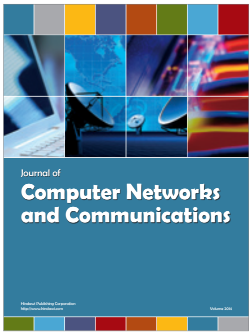

International Journal of
Distributed
Sensor Networks

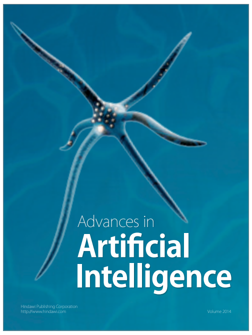

Journal of
Industrial Engineering

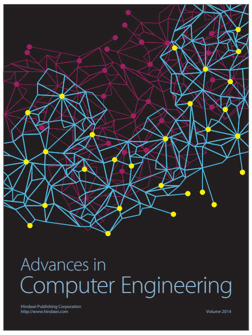

Applied
Computational
Intelligence and Soft
Computing


Advances in
Fuzzy
Systems

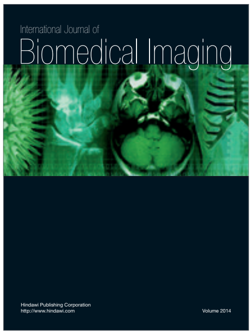

Modelling &
Simulation
in Engineering

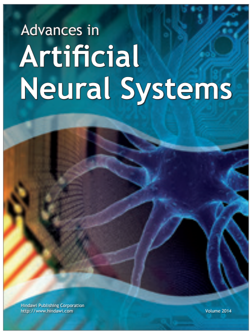

Journal of
Computer Networks
and Communications

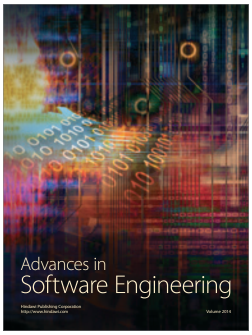

Advances in
Artificial
Intelligence


Advances in
Computer Engineering

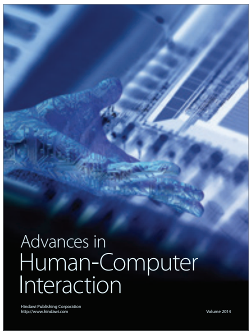
International Journal of
Computer Games
Technology

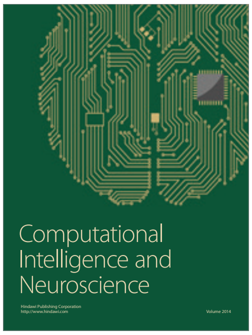

International Journal of
Biomedical Imaging


Advances in
Artificial
Neural Systems


Advances in
Software Engineering

Journal of
Robotics


Advances in
Human-Computer
Interaction


Computational
Intelligence and
Neuroscience

International Journal of
Reconfigurable
Computing

Journal of
Electrical and Computer
Engineering