

Research Article

Color-Texture-Based Image Retrieval System Using Gaussian Markov Random Field Model

**Meng-Hsiun Tsai,¹ Yung-Kuan Chan,¹ Jiun-Shiang Wang,²
Shu-Wei Guo,² and Jiunn-Lin Wu²**

¹ *Department of Management Information Systems, National Chung Hsing University,
402 Taichung, Taiwan*

² *Department of Computer Science and Engineering, National Chung Hsing University,
402 Taichung, Taiwan*

Correspondence should be addressed to Yung-Kuan Chan, ykchan@nchu.edu.tw

Received 23 April 2009; Revised 10 August 2009; Accepted 5 November 2009

Recommended by Panos Liatsis

The techniques of K -means algorithm and Gaussian Markov random field model are integrated to provide a Gaussian Markov random field model (GMRFM) feature which can describe the texture information of different pixel colors in an image. Based on this feature, an image retrieval method is also provided to seek the database images most similar to a given query image. In this paper, a genetic-based parameter detector is presented to decide the fittest parameters used by the proposed image retrieval method, as well. The experimental results manifested that the image retrieval method is insensitive to the rotation, translation, distortion, noise, scale, hue, light, and contrast variations, especially distortion, hue, and contrast variations.

Copyright © 2009 Meng-Hsiun Tsai et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

Much attention has been devoted to the design of image databases over the past few years [1–8]. Image retrieval is an important task in many image database applications, such as office automation, medical image archiving, digital library, multimedia publishing, computer-aided design, and geographic information systems.

Most traditional and common methods of image retrieval utilize some method of adding metadata such as captioning, keywords, or descriptions to the images so that retrieval can be performed over the annotation words [4]. As these systems are built upon a large database, the textual-feature-based method becomes not only cumbersome but also inadequate to represent image contents. Hence, many content-based image retrieval (CBIR) systems have been proposed in the related literatures [1–8]. The CBIR aims at avoiding

the use of textual descriptions and instead retrieves images based on their visual similarity to a user-supplied query image or user-specified image features.

Low-level features, such as colors [9, 10], shapes [7, 8], and textures [11, 12], are extensively used to index the image features of the CBIR systems. The shape-based image retrieval system searches for images containing the objects similar to the objects specified by a query image. However, locating and recognizing objects from images is a real challenge. One of the difficulties is to separate the objects from background. These difficulties come from discretization, occlusions, poor contrast, viewing conditions, noise, complicated objects, complicated background, and so forth. Moreover, the shape-based image retrieval method can deal with only the images with simple object shapes efficiently. For complex object shapes, the region-based method has to build a binary sequence by using smaller grid cells so that more accurate results can be obtained; nevertheless, the storage of indices and retrieval time may increase tremendously.

The color and texture attributes have been very successfully used in retrieving images with similar feature distributions. The color-based image retrieval method, such as the conventional color histogram (CCH) [13] and the fuzzy color histogram (FCH) [14], measures the similarity of two images with their distance in color space. The extraction of the color features follows a similar progression in each of the four methods: (1) selection of the color space, (2) quantization of the color space, (3) extraction of the color feature, (4) derivation of an appropriate distance function [13]. Color attribute may avoid object identification and extraction in image retrieval. Color may provide multiple measurements at a single pixel of the image, and often enable the classification to be done without complex object segmentation. Vu et al. [7] proposed a color-based image retrieval method, called a SamMatch method, to process the query by sampling-based matching approach. The SamMatch method has the benefits of region-based techniques without reliance on the highly variable accuracy of segmentation methods. However, this method requires the user to point out the object areas at the time of the query.

Unfortunately, color-based image retrieval systems often fail to retrieve the images taken from the same scene but under different time or conditions, for example, the images of a countryside taken at dawn, dusk, or noon under a clear or a cloudy sky. In another scenario where a same scene is imaged by different devices, using one image taken by one device as the query example may fail to find the same scene taken by other devices.

Texture attribute depicts the "surface" of an object. Intuitively, this term refers to properties such as smoothness, coarseness, and regularity of an object. Generally, the structural homogeneity does not come from the presence of a single color or intensity, but requires the interaction of various intensities within a region. Texture similarity is often useful in distinguishing objects with similar colors, such as sky and sea as well as leaves and grass. Making texture analysis is a real challenge. One of the ways to perform content-based image retrieval using texture as the cue is to segment an image into a number of different texture regions and perform a texture analysis on each region. However, segmentation can sometimes be problematic for image retrieval. In addition, texture is quite difficult to describe and subject to the difference of human perception. No satisfactory quantitative definition of texture is available at this time.

The notion of texture appears to depend upon three ingredients [15]. First, some local "order" is repeated over a region which is large in comparison to the order's size. Second, the order consists in the nonrandom arrangement of elementary parts. Third, there are roughly uniform entities having approximately the same dimension everywhere within the textured region. Liu and Picard [16], and Niblack et al. [17, 18] used contrast, coarseness

(5, 2)	(4, 3)	(3, 2)	(4, 2)	(5, 1)
(4, 4)	(2, 2)	(1, 2)	(2, 1)	(4, 1)
(3, 1)	(1, 1)	(0, 0)	(1, 1)	(3, 1)
(4, 1)	(2, 1)	(1, 2)	(2, 2)	(4, 4)
(5, 1)	(4, 2)	(3, 2)	(4, 3)	(5, 2)

Figure 1: The neighborhood of $x_{0,0}$ with a block of 5×5 pixels.

and directionality models to achieve texture classification and recognition. Huang and Dai [19] proposed a texture-based image retrieval method. The method associates one coarse and one fine feature descriptors with each image. Both descriptors are derived from the coefficients of wavelet transform of the original image. The coarse feature descriptor is used at the first stage to quickly screen out nonpromising images; the fine feature descriptor is subsequently employed to find the truly matched images. However, Huang's method cannot give a high accuracy rate of precisely seeking the desired image.

To solve the problems mentioned above, this paper proposes a color-texture-based image retrieval system (CTBIR method). The CTBIR method integrates the color and texture attributes of an image. It employs K -means algorithm [20] to classify the colors in all database images into clusters, then calculates the average of the pixel colors in each cluster, and uses this average color to represent the colors of the pixels in the cluster. Next, the CTBIR method utilizes Gaussian Markov random field model (GMRFM) to analyze the texture of each representative color in an image. A genetic algorithm is also provided to decide the most suitable parameters used in the CTBIR method. Besides, the performance of the CTBIR method is investigated by experiments. The effects of various image variations, such as rotation, translation, distortion, noise, scale, light, hue, and contrast on the performance of the CTBIR method are explored as well.

2. Related Works

This section will give a brief introduction of Gaussian Markov Random Field Model, genetic algorithm, and other techniques to be used in the CTBIR method.

2.1. Gaussian Markov Random Field Model

Texture plays a very significant role in analyzing image characteristics. Chellappa and Chatterjee [11] adopted GMRFM to describe the relationship between each pixel and its surrounding pixels. Generally, there exist some similarities and correlations in gray-level variance between adjacent pixels. In GMRFM, the varying degree of texture in images is statistically calculated mainly on the basis of the property. While Chellappa and Chatterjee used GMRFM to describe the textures of objects on an gray-scale image and segment the objects from the image, in this paper it will be used to characterize the textures of images in order to retrieve the images from a database which are most similar to the given query image.

In GMRFM [12, 21–23], all the pixels in each $m_G \times m_G$ block B are classified according to their relationship with the central pixel $x_{0,0}$. Figure 1 illustrates an example of GMRFM where $m_G = 5$ and (i, j) indicates the neighboring pixels $x_{i,j}$ of $x_{0,0}$, where $x_{i,j}$ is the pixel belonging

to the j th category in the i th rank relative to $x_{0,0}$. A set of eigenvalues $\theta_{i,j}$'s are employed to describe the relationship between each $x_{i,j}$ and $x_{0,0}$ where $x_{0,0}$ is calculated as [11]

$$x_{0,0} = \sum_{(i,j) \in N_{x_{0,0}}} \theta_{i,j} x_{i,j} + e. \quad (2.1)$$

In this formula, e indicates a Gaussian distribution of noises. $N_{x_{0,0}}$ denotes the neighbors of $x_{0,0}$ shown in Figure 1. Let X and θ be the column vectors composed, respectively, of $x_{i,j}$ and $\theta_{i,j}$ excluding $x_{0,0}$ and $\theta_{0,0}$. Then, one can calculate θ as follows:

$$\theta = \left[\sum XX^T \right]^{-1} \left[\sum x_{0,0} X \right], \quad (2.2)$$

which can be used to depict the texture of B . The larger most of the values in vector θ is, the rougher B is; otherwise, B is smoother. Yue [8] also uses θ to describe the texture of a telemeter image of glacier and divides the image into some regions according to their textures.

2.2. Genetic Algorithm

Seo [24] applied a genetic algorithm for selection and feature combination in pattern recognition applications. Ballerini et al. [25] used a genetic algorithm to select a subset of texture features from all the features extracted. Lai and Chen [26] used the interactive genetic algorithm to tune human judgment results on similarity of images. Joshi and Tapaswi [2] applied a genetic algorithm to decide the most plausible matching between two images in image retrieval. Stejić et al. [6] proposed an image retrieval system using local similarity patterns. The system used a genetic algorithm to find an optimal assignment of similarity criteria to image regions. Chan and King [1] provided a trademark image retrieval system which employed a genetic algorithm to find the weighting factors in the dissimilarity function by integrating five shape features. This paper will adopt a genetic algorithm to decide the fittest parameters which will be used by the CTBIR method.

Genetic algorithm (GA) [27] is a heuristic optimization method that operates through a determined and randomized search. The set of possible solutions for an optimization problem is considered as a population of individuals, and the degree of adaptation of an individual to its environment is specified by its fitness. A chromosome, essentially a set of character strings, represents the coordinate of an individual in the search space. A gene is a subsection of a chromosome that encodes the value of a single parameter that is being optimized. Typical encoding for a gene can be binary or integer.

Derived from the biological model of evolution, genetic algorithms operate on the Darwinian principle of natural selection, which holds that, given a certain population, only the individuals that adapt well to their environment can survive and transmit their characteristics to their descendants. A genetic algorithm consists of three major operations: selection, crossover, and mutation. Selection evaluates all individuals, and only those best fitted to their environment survive. Crossover recombines the genetic material of two individuals to form new combinations with the potential for better performance. Mutation induces changes in a small number of chromosomal units with the goal of maintaining sufficient population diversity during the optimization process.

2.3. System Evaluation

Whenever a certain image f_q is selected as a query image, the image retrieval system delivers the user R database images whose matching distances to f_q are shortest. If the desired database image is one of these R transmitted images, we say the system correctly finds out the desired image [9, 10]. Otherwise, the system fails to respond to the desired image. In the following experiments, the accuracy rate of querying for a system will be explained with ACC.

The other evaluation, normalized modified retrieval rank (NMRR), proposed by MPEG-7 [28] is usually used as a benchmark for system evaluation. NMRR not only indicates how many of the correct items are retrieved, but also how highly they are ranked among the retrieved items. NMRR is defined by

$$\text{NMRR}(q) = \frac{\left(\sum_{k=1}^{\text{NG}(q)} (\text{Rank}(k)/\text{NG}(q))\right) - 0.5 - (\text{NG}(q)/2)}{K(q) + 0.5 - 0.5 \times \text{NG}(q)}, \quad (2.3)$$

where $\text{NG}(q)$ is the size of the ground truth set for a query image q , $\text{Rank}(k)$ is the ranking of the ground truth images by the retrieval algorithm, and $K(q)$ specifies the “relevance rank” for each query. As the size of the ground truth set is normally unequal, a suitable $K(q)$ is determined by

$$K(q) = \min(4 \times \text{NG}(q), 2 \times \text{GTM}), \quad (2.4)$$

where GTM is the maximum of $\text{NG}(q)$ for all queries. If $\text{Rank}(k) > K(q)$, then $\text{Rank}(k)$ is changed into $K(q) + 1$. The NMRR is in the range of $[0, 1]$, and the smaller NMRR is, the better the retrieval performance will be. ANMRR is defined as the average NMRR over a range of queries, and is given by

$$\text{ANMRR} = \frac{1}{\text{NQ}} \sum_{q=1}^{\text{NQ}} \text{NMRR}(q), \quad (2.5)$$

where NQ is the number of query images.

3. Color-Texture-Based Image Retrieval System

This paper proposes a color-texture-based image retrieval system (CTBIR method) on the basis of the color and texture attributes of images. The system contains two phases—feature extraction and image matching. The feature extraction phase is to extract the GMRFM feature of an image. The image matching phase is to compare the GMRFM features of the given query image and each database image and then, deliver the most similar database image relative to the query image to the user.

3.1. Feature Extraction

For large sized blocks, the GMRFM feature needs high feature dimensions to precisely describe the texture of an image. The CTBIR method will require much memory space to hold the feature of the image and much time to compare the feature with that of each database image. In this paper, every gray-scale image is hence divided into many overlapping blocks of 4×4 pixels, as shown in Figure 2(a), and the relationship of the positions for the 16 pixels in a block is depicted in Figure 2(b). Afterwards, each block B of 4×4 pixels is further partitioned into four overlapping subblocks, $B_{1,1}$, $B_{1,2}$, $B_{2,1}$, and $B_{2,2}$, of 3×3 pixels where each subblock $B_{i,j}$ includes nine pixels: $x_{i-1,j-1}$, $x_{i-1,j}$, $x_{i-1,j+1}$, $x_{i,j-1}$, $x_{i,j}$, $x_{i,j+1}$, $x_{i+1,j-1}$, $x_{i+1,j}$, $x_{i+1,j+1}$. Let $\theta_{i,j}$ in the GMRFM θ be the j th category in the i th rank in B . $\theta_{i,j}$ is defined as follows:

$$\theta_{i,j} = \frac{\sum_{h=1}^2 \sum_{k=1}^2 \theta_{h,k}^{B_{i,j}}}{4}, \quad (3.1)$$

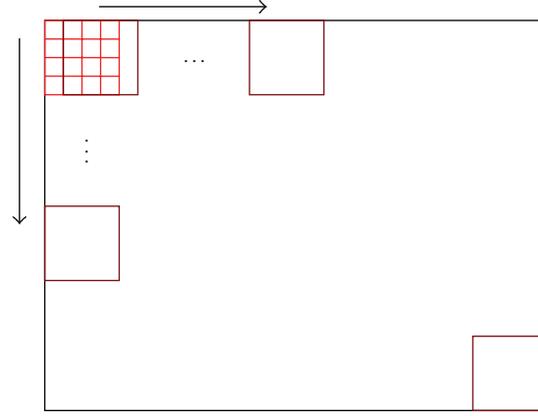
where $\theta_{h,k}^{B_{i,j}}$ for $h, k = \{1, 2\}$, are the texture estimates and are computed as follows:

$$\begin{aligned} \theta_{1,1}^{B_{i,j}} &= \frac{x_{i,j} \times (x_{i,j-1} + x_{i,j+1})}{2 \times \left(\left(\sum_{h=-1}^1 \sum_{k=-1}^1 x_{i+h,j+k}^2 \right) - x_{i,j}^2 \right)}, \\ \theta_{1,2}^{B_{i,j}} &= \frac{x_{i,j} \times (x_{i-1,j} + x_{i+1,j})}{2 \times \left(\left(\sum_{h=-1}^1 \sum_{k=-1}^1 x_{i+h,j+k}^2 \right) - x_{i,j}^2 \right)}, \\ \theta_{2,1}^{B_{i,j}} &= \frac{x_{i,j} \times (x_{i-1,j+1} + x_{i+1,j-1})}{2 \times \left(\left(\sum_{h=-1}^1 \sum_{k=-1}^1 x_{i+h,j+k}^2 \right) - x_{i,j}^2 \right)}, \\ \theta_{2,2}^{B_{i,j}} &= \frac{x_{i,j} \times (x_{i-1,j-1} + x_{i+1,j+1})}{2 \times \left(\left(\sum_{h=-1}^1 \sum_{k=-1}^1 x_{i+h,j+k}^2 \right) - x_{i,j}^2 \right)}. \end{aligned} \quad (3.2)$$

As a result, each block B comprises four feature values $\theta_{1,1}$, $\theta_{1,2}$, $\theta_{2,1}$, and $\theta_{2,2}$.

Each pixel in a full color image I consists of three color components: R, G, and B. The CTBIR method uses the three color components to construct three gray-scale images of the same size as I : I_r , I_g , and I_b , and each of which is entirely composed of R, G, and B color components from I , respectively. Next, I , I_r , I_g , and I_b are all divided into overlapping blocks of 4×4 pixels and the attributes θ 's in each block with regard to I_r , I_g , and I_b are calculated, which are, respectively, notated as $\theta_{1,1}^r$, $\theta_{1,2}^r$, $\theta_{2,1}^r$, $\theta_{2,2}^r$, $\theta_{1,1}^g$, $\theta_{1,2}^g$, $\theta_{2,1}^g$, $\theta_{2,2}^g$, $\theta_{1,1}^b$, $\theta_{1,2}^b$, $\theta_{2,1}^b$, $\theta_{2,2}^b$.

In a full color image, a pixel is generally described by a 24-bit memory space; there are 2^{24} possible colors in an image. However, it is impossible for most images to use all of the possible colors. In order to reduce the memory space required for storing image features and querying time, the CTBIR method employs K -means algorithm to group the pixels in all database images into K groups according to their colors. Moreover, the average color of all the pixels in one group is considered to be a representative color of all images (including database images and query image). The K representative colors comprise a common palette (CP).


 (a) The way to cut overlapped blocks of 4×4

$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$x_{1,4}$
$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$x_{2,4}$
$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	$x_{3,4}$
$x_{4,1}$	$x_{4,2}$	$x_{4,3}$	$x_{4,4}$

(b) The relationship of positions for pixels in a block

Figure 2: The cutting way for GMRFM feature and the relationship of positions.

In every color image I , the i th color C_i in CP is given a set of variables: $\theta_{1,1}^{CP_{i,r}}, \theta_{1,2}^{CP_{i,r}}, \theta_{2,1}^{CP_{i,r}}, \theta_{2,2}^{CP_{i,r}}, \theta_{1,1}^{CP_{i,g}}, \theta_{1,2}^{CP_{i,g}}, \theta_{2,1}^{CP_{i,g}}, \theta_{2,2}^{CP_{i,g}}, \theta_{1,1}^{CP_{i,b}}, \theta_{1,2}^{CP_{i,b}}, \theta_{2,1}^{CP_{i,b}}, \theta_{2,2}^{CP_{i,b}}$, and A_i . For each block B of 4×4 pixels in I , the average C_a of all the pixel colors in B is calculated. If C_a is similar to C_i in CP, the blocks $\theta_{1,1}^r, \theta_{1,2}^r, \theta_{2,1}^r, \theta_{2,2}^r, \theta_{1,1}^g, \theta_{1,2}^g, \theta_{2,1}^g, \theta_{2,2}^g, \theta_{1,1}^b, \theta_{1,2}^b, \theta_{2,1}^b, \theta_{2,2}^b$ are, respectively, added to $\theta_{1,1}^{CP_{i,r}}, \theta_{1,2}^{CP_{i,r}}, \theta_{2,1}^{CP_{i,r}}, \theta_{2,2}^{CP_{i,r}}, \theta_{1,1}^{CP_{i,g}}, \theta_{1,2}^{CP_{i,g}}, \theta_{2,1}^{CP_{i,g}}, \theta_{2,2}^{CP_{i,g}}, \theta_{1,1}^{CP_{i,b}}, \theta_{1,2}^{CP_{i,b}}, \theta_{2,1}^{CP_{i,b}}, \theta_{2,2}^{CP_{i,b}}$, as well as A_i is increased by 1. Then, the CTBIR method adopts these θ 's and A 's of all colors in CP as the feature of I , where we call it the GMRFM feature of I . The GMRFM feature of I hence contains $12 \times K$ of θ 's values and K of A 's values, where A_i is the number of 4×4 blocks whose average pixel color is similar to one color C_i of CP in I . Hence, A 's can be used to describe the distribution of pixels colors in I , while the textures can be described by means of θ 's. On the whole, the distribution of the textures for different colors in I can be explained by A 's and θ 's.

The GMRF model assumes that the texture field is stochastic, stationary and satisfies a conditional independence assumption. It is able to capture the local (spatial) contextual variations in an image and is effective for the analysis and synthesis of random looking textures. For each color image I , the CTBIR method generates a set of $\theta_{i,j}^{CP_{k,r}}, \theta_{i,j}^{CP_{k,g}}, \theta_{i,j}^{CP_{k,b}}$, and A_i for $i, j = \{1, 2\}$ and $1 \leq k \leq K$. Here, (i, j) illustrates the orientation of GMRFM feature of I ; $CP_{k,r}$, $CP_{k,g}$, and $CP_{k,b}$ are the three color components of the k th color in CP. A_i describes the color distribution of I . Hence, the GMRFM feature also can describe the texture orientation and color information of I .

3.2. Image Matching

Minkowski distance is widely used to calculate the distance between two multidimension vectors [5]. Suppose that (x_1, x_2, \dots, x_K) and $(x'_1, x'_2, \dots, x'_K)$ are two K -dimension vectors. The Minkowski distance D_r of the two vectors can be defined by

$$D_r = \left(\sum_{i=1}^K |x_i - x'_i|^r \right)^{1/r}, \quad (3.3)$$

where r is a user-defined constant. D_r is a Manhattan distance if $r = 1$ and while D_r is a Euclidean distance in case of $r = 2$.

Let $\theta_{i,j}^{CP_{k,C}}$ and $\theta'_{i,j}^{CP_{k,C}}$, respectively, be the $\theta_{i,j}^{CP_C}$ of I and I' with regard to the k th color in CP, where $C = r, g, \text{ or } b$; i and j are 1 or 2 while A_k and A'_k stand for A , respectively, of I and I' with regard to the k th color in CP. The image matching distance D between I and I' is defined as follows:

$$D = \sum_{k=1}^K \sum_{C=r,g,b} |A_k - A'_k|^{r_1} \left(\sum_{i=1}^2 \sum_{j=1}^2 \left| \theta_{i,j}^{CP_{k,C}} - \theta'_{i,j}^{CP_{k,C}} \right|^{r_2} \right)^{1/r_2}, \quad (3.4)$$

where r_1 and r_2 are two user-defined constants.

4. Genetic Algorithm-Based Parameter Detector

The performance of CTBIR method is significantly affected by r_1 and r_2 . In this paper, a genetic algorithm-based parameter detector (GBPD) is used to determine the most suitable r_1 and r_2 . The GBPD makes use of a binary string composed of $m_1 + m_2$ binary bits to represent a chromosome Ch in which the first m_1 bits and the remaining m_2 bits are used to describe r_1 and r_2 , respectively. For each chromosome Ch , r_1 and r_2 are encoded as $r_1 = \text{gap}_1 \times n_1$ and $r_2 = \text{gap}_2 \times n_2$, where n_1 is the number of 1 bit in the first m_1 bits of Ch , and n_2 is the number of 1 bit in the remaining m_2 bits of Ch ; gap_1 and gap_2 are the maximum estimated errors of r_1 and r_2 .

For a certain application, one can accumulate some of its historic images (including database images and query images). After that, he can apply the accumulated historic images to train the most appropriate values for r_1 and r_2 via the genetic algorithm. In this genetic algorithm, we define the accuracy rate (ACC) obtained by the CTBIR method based on the accumulated historic images and the values of r_1 and r_2 encoded by a chromosome as the fitness of the chromosome.

GBPD first randomly generates N chromosomes, each with $m_1 + m_2$ binary bits. To evolve the best solution, the genetic algorithm repeatedly executes mutation, crossover, and selection operations until the relative fitness of the reserved chromosomes is very similar.

In the mutation operation, for each of the N reserved chromosomes, the GBPD uses a random number generator to specify one bit b_1 from the first m_1 bits and the other bit b_2 from the remaining m_2 bits in the chromosome. Later, b_1 and b_2 are replaced by $\neg b_1$ and $\neg b_2$ to generate a new chromosome, where \neg signifies the operator "NOT."

In the crossover operation, GBPD similarly uses a random number generator to designate N' pairs of chromosomes from the N reserved chromosomes. Let $Ch[i \cdots j]$ be the substring consisting of the i th to j th bits in Ch . For each chromosome pair (Ch_1, Ch_2) , the genetic algorithm concatenates

$$\begin{aligned} Ch_1 \left[1 \cdots \left\lfloor \frac{m_1}{2} \right\rfloor \right], \quad Ch_2 \left[\left(\left\lfloor \frac{m_1}{2} \right\rfloor + 1 \right) \cdots m_1 \right], \quad Ch_1 \left[(m_1 + 1) \cdots \left(m_1 + \left\lfloor \frac{m_2}{2} \right\rfloor \right) \right], \\ Ch_2 \left[\left(m_1 + \left\lfloor \frac{m_1}{2} \right\rfloor + 1 \right) \cdots (m_1 + m_2) \right] \end{aligned} \quad (4.1)$$

into a new chromosome, and

$$\begin{aligned} Ch_2 \left[1 \cdots \left\lfloor \frac{m_1}{2} \right\rfloor \right], \quad Ch_1 \left[\left(\left\lfloor \frac{m_1}{2} \right\rfloor + 1 \right) \cdots m_1 \right], \quad Ch_2 \left[(m_1 + 1) \cdots \left(m_1 + \left\lfloor \frac{m_2}{2} \right\rfloor \right) \right], \\ Ch_1 \left[\left(m_1 + \left\lfloor \frac{m_1}{2} \right\rfloor + 1 \right) \cdots (m_1 + m_2) \right] \end{aligned} \quad (4.2)$$

into other new chromosome.

In the selection operation, N optimal chromosomes are selected from the N chromosomes reserved in the previous iteration along with the N and $2 \times N'$ chromosomes created in the mutation and crossover operations according to their fitness. GBPD continuously performs the mutation, crossover, and selection operations, until the related fitness of the reserved N chromosomes is very close or the number of iterations is equal to the specified maximum number of generations (in this paper, the maximum number of generations is set to be 100). Finally, GBPD multiplies gap_1 (resp., gap_2) by the numbers of 1-bits in $Ch[1 \cdots m_1]$ (resp., $Ch[m_1 + 1 \cdots m_1 + m_2]$) to get r_1 (resp., r_2), where Ch is the chromosome having the best fitness in the N reserved chromosomes.

Let $m_1 = m_2 = 14$, $gap_1 = gap_2 = 0.1$, and let Figure 3(a) be a chromosome Ch . Derived from Ch , $r_1 = 0.1 \times 4 = 0.4$, and $r_2 = 0.1 \times 6 = 0.6$. Figure 3(b) demonstrates a new chromosome created from Ch by the mutation operator, where the underlined bits are the randomly selected b_1 and b_2 . Figure 3(c) displays two new chromosomes Ch'_1 and Ch'_2 , generated from the two chromosomes Ch_1 and Ch_2 through the crossover operator.

5. Experiments

The purpose of this section is to evaluate the performances of the CTBIR method by experiments. Let $SetD = \{f_1^d, f_2^d, f_3^d, \dots, f_{1087}^d\}$ and $SetQ = \{f_1^q, f_2^q, f_3^q, \dots, f_{1087}^q\}$ be two image sets, each of which contains 1087 full color images. The images in $SetD$ are employed as the database images and those in $SetQ$ are used as the query images. Some of them are captured from animations, where each image pair (f_i^d, f_i^q) is randomly picked from a same animation. Most of the animations were downloaded from <http://www.mcsh.kh.edu.tw>. The rest of the images was scanned from natural images and trademark pictures.

In $SetD$ and $SetQ$, the former 500 images are almost cartoon images and the remainders are natural ones. Figure 4(a) shows parts of the query and database images in the former 500 ones of $SetD$ and $SetQ$, and Figure 4(b) demonstrates parts of those in the latter 587 images of $SetD$ and $SetQ$. In the following experiments, the parameters m_1 , m_2 , N , N' , gap_1 , and gap_2

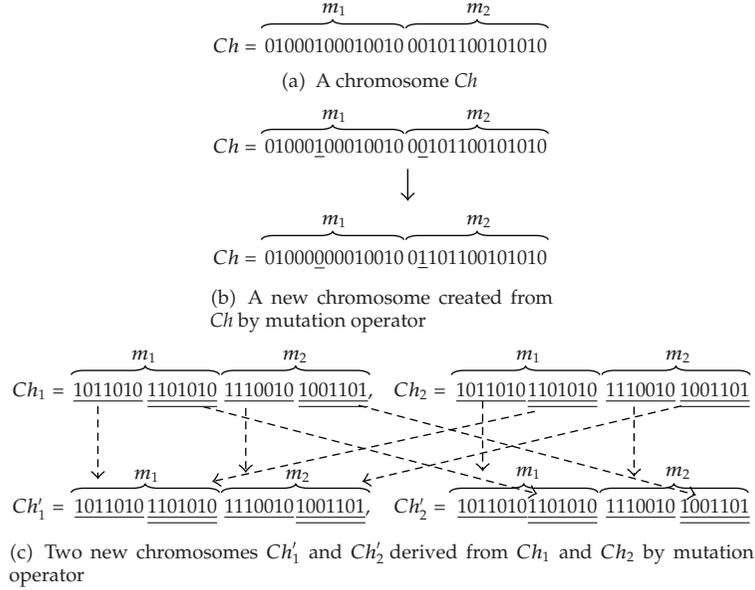


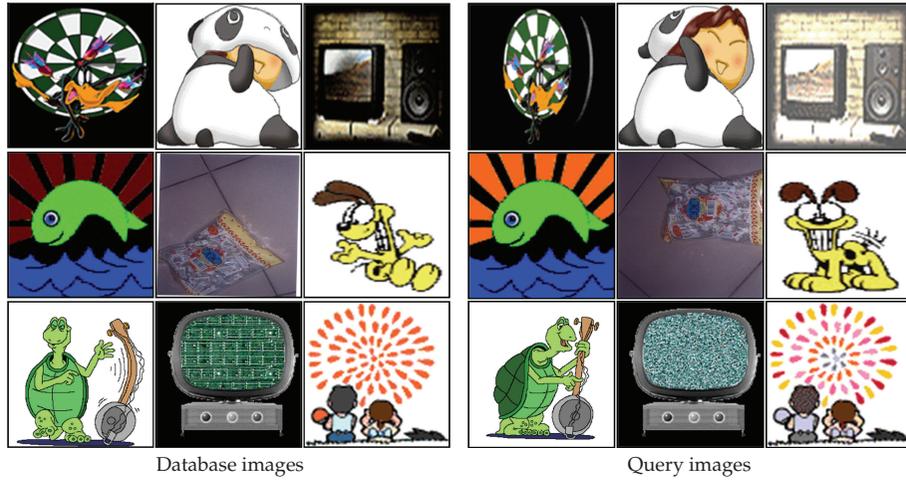
Figure 3: An example of GBP.

are given to be 40, 40, 20, 10, 0.05, and 0.05 to determine the most suitable r_1 and r_2 via the genetic algorithm. Moreover, the size K of CP is set to 16.

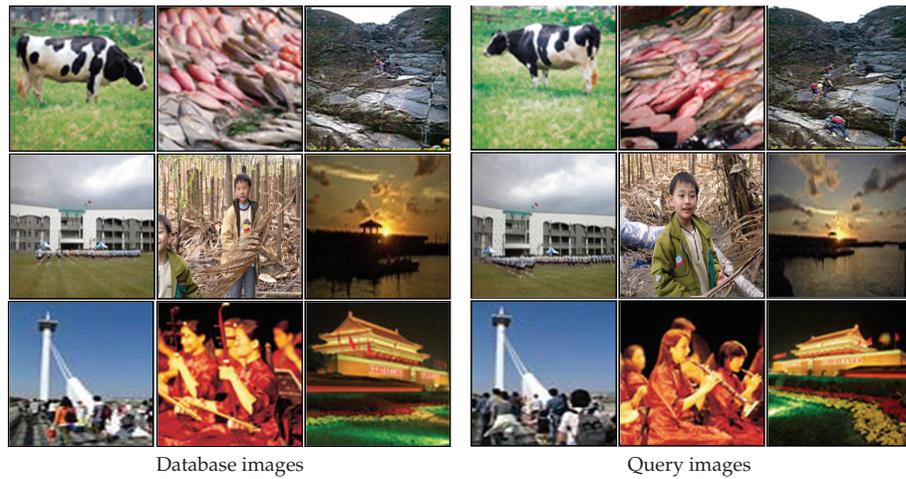
The first experiment is to investigate the performance of the CTBIR method in retrieving cartoon images. First, the images $\{f_1^d, f_2^d, \dots, f_{100}^d\}$ and $\{f_1^q, f_2^q, \dots, f_{100}^q\}$ are, respectively, specified to be the database images and the query images, and the GBP is applied to determine the most suitable r_1 and r_2 . The experimental results obtained by the GBP tell that the most suitable r_1 and r_2 are 1.60 and 0.40. Then, the images $\{f_1^d, f_2^d, \dots, f_{500}^d\}$ and $\{f_1^q, f_2^q, \dots, f_{500}^q\}$ are used as the database images and the query images based on the most suitable $r_1 = 1.60$ and $r_2 = 0.40$.

The second experiment is to scrutinize the performance of the CTBIR method in retrieving natural images. In this experiment, $\{f_{501}^d, f_{502}^d, \dots, f_{600}^d\}$ and $\{f_{501}^q, f_{502}^q, \dots, f_{600}^q\}$ are first taken as the database images and the query images, respectively, and the GBP is employed to find out the most suitable $r_1 = 1.35$ and $r_2 = 0.85$. In the wake, $\{f_{501}^d, f_{502}^d, \dots, f_{1087}^d\}$ and $\{f_{501}^q, f_{502}^q, \dots, f_{1087}^q\}$ are used as the database images and the query images, and the CTBIR method is assigned to retrieve the expected database images with $r_1 = 1.35$ and $r_2 = 0.85$. Figure 5 shows the obtained r_1 , and r_2 , in 100 iterations obtained by the GBP in experiments 1 and 2. This experimental result explains that a bigger r_1 is required to describe the cartoon images, since the color distributions between most natural images are more varied compared to those between most cartoon images even though the colors of both images are reduced to reserve only a few principal colors.

In the third experiment, the images $\{f_1^d, f_2^d, \dots, f_{100}^d, f_{501}^d, f_{502}^d, \dots, f_{600}^d\}$ and $\{f_1^q, f_2^q, \dots, f_{100}^q, f_{501}^q, f_{502}^q, \dots, f_{600}^q\}$ are considered as the database images and the query images for determining the most suitable $r_1 = 1.65$ and $r_2 = 0.50$ by using the GBP. Then, the images in $SetD = \{f_1^d, f_2^d, f_3^d, \dots, f_{1087}^d\}$ and $SetQ = \{f_1^q, f_2^q, f_3^q, \dots, f_{1087}^q\}$ are used as the database images and the query images based on the most suitable $r_1 = 1.65$ and $r_2 = 0.50$. Table 1 displays the results of the first to third experiments.



(a) Partial cartoon images



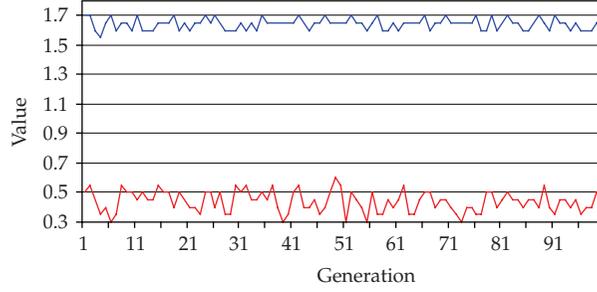
(b) Partial natural images

Figure 4: Partial testing images.

Table 1: The results of the first to third experiments.

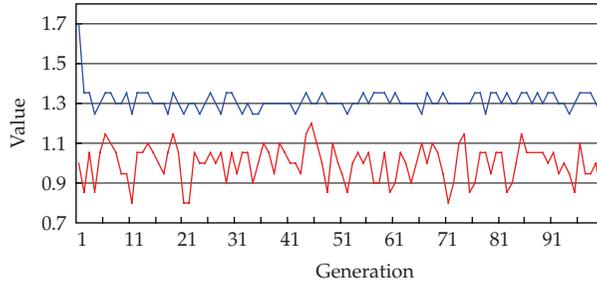
Experiment	Images	R_1	R_2	ACC ($R=1$) (%)	ACC ($R=10$) (%)	ANMRR
First	Previous 500 images in $SetD$ and $SetQ$	1.60	0.40	91.5	96.8	0.079
Second	Later 587 images in $SetD$ and $SetQ$	1.35	0.85	77.8	91.8	0.231
Third	All images in $SetD$ and $SetQ$	1.65	0.50	82.4	93.2	0.161

The fourth experiment is to compare the performances of the CTBIR, Huang's and SamMatch methods by using the images in $SetD = \{f_1^d, f_2^d, f_3^d, \dots, f_{1087}^d\}$ and $SetQ = \{f_1^q, f_2^q, f_3^q, \dots, f_{1087}^q\}$. Table 2 demonstrates the experimental results where r_1 and r_2 are given



— r_1
— r_2

(a) Cartoon images



— r_1
— r_2

(b) Nature images

Figure 5: The r_1 and r_2 obtained by the GBPD in 100 iterations.**Table 2:** The performances (ANMRR) obtained in Experiment 3.

Testing images	Method		
	CTBIR	Huang's	SamMatch
Previous 500 image pairs in <i>SetD</i> and <i>SetQ</i>	0.079	0.137	0.229
Later 587 image pairs in <i>SetD</i> and <i>SetQ</i>	0.231	0.242	0.574
All Images in <i>SetD</i> and <i>SetQ</i>	0.161	0.194	0.415

to be 1.65 and 0.50. The experimental results indicate that the CTBIR method gives much better ANMRR than Huang's and SamMatch methods, especially for cartoon images. The CTBIR method totally spends 110.4 seconds to run the 1087 queries in this experiment. In this experiment, CP consists of $K = 16$ colors each of which corresponds to twelve θ 's and one A ; each value is held in a 4-byte memory space. Hence, the CTBIR method takes $1087 \times 16 \times 13 \times 4 = 904,384$ bytes to hold the whole GMRFM features of the 1087 database images.

The SamMatch method resizes each image into a 256×256 image, reduces all pixel colors in the image into only 256 colors, and divides the image into 16×16 blocks and uses the average colors of the blocks as the feature of the image. Hence, 113 samples evenly spread out in each image, and each dimension is held by a 4-byte memory space. It consumes 491.324 ($= 1087 \times 113 \times 4$) bytes to hold the features of all database images. In addition, the SamMatch

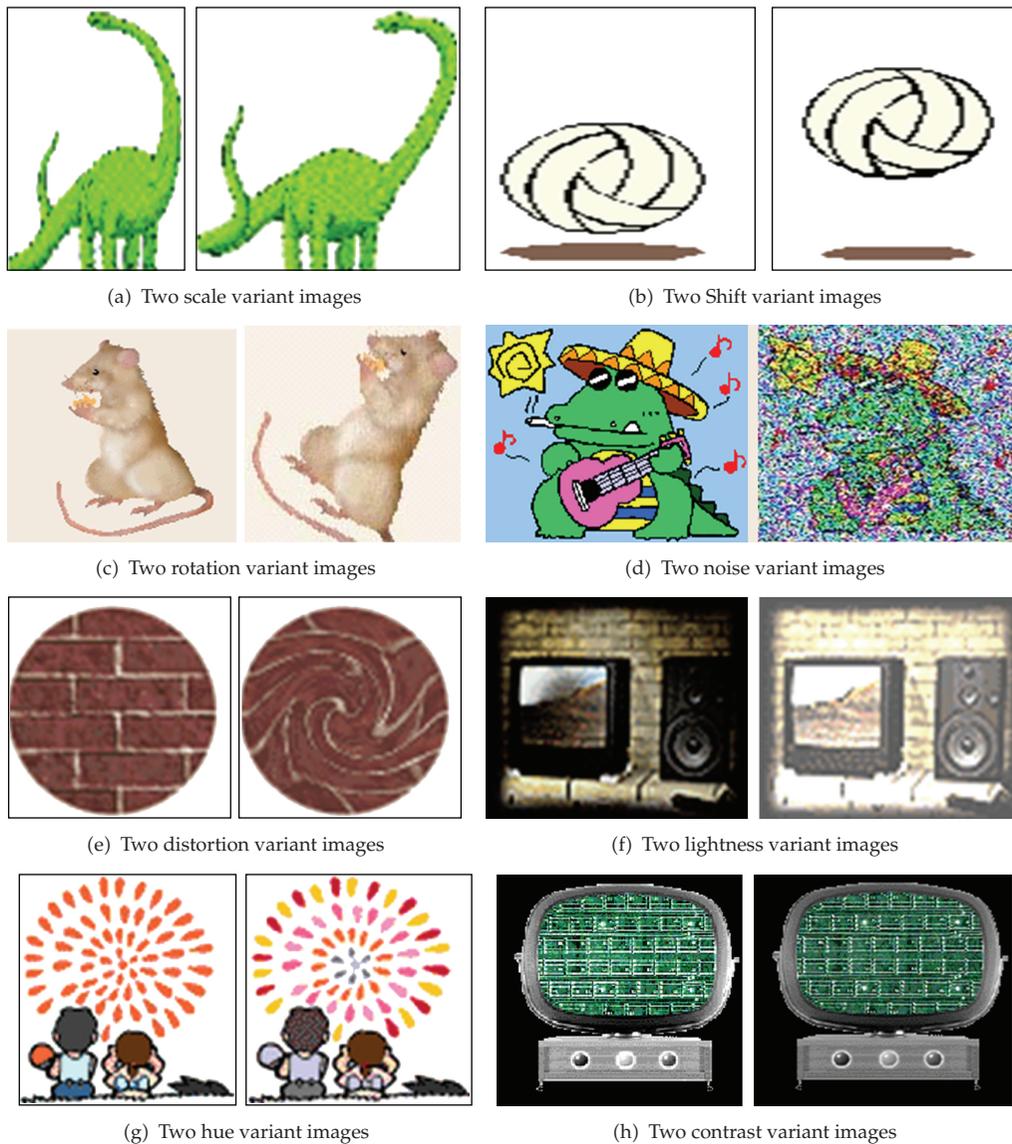


Figure 6: Some variant image pairs.

method takes 245.02 seconds to execute the 1087 queries. The Huang's method consumes 620,928 bytes memory space to save the features of 1087 database images, and spends 125.6 seconds in running the 1087 queries.

When one takes some pictures, the lens may be adjusted to different positions or different directions. Moving the position of certain objects or pixels on an image is called the shift variation of the image, rotating the objects in certain degree is called the rotation variation of the image, and being added some noises in the image is called the noise variation of the image. An image may be enlarged or reduced because of different camera resolution setups. We call this phenomenon a scale variation of the image. In the real world, the

	$I = 11$	$I = 25$	$I = 35$	$I = 49$	$I = 100$
$I_{q,i}$					
$I_{r,i}$					
$I_{d,i}$					
$I_{n,i}$					
$I_{s,i}$					
$I_{h,i}$					
$I_{l,i}$					
$I_{c,i}$					

Figure 7: The i th images in sets $S_q, S_r, S_d, S_n, S_s, S_h, S_l$, and S_c .

distortion variation is a very common phenomenon too. For example, putting a picture on an uneven plane may deform the objects on the picture. Besides that, the hue, contrast, and light variations, such as the image pairs in Figure 6, may be often generated. These variations may make the image regarded as a different image from the original one by an image retrieval system. An excellent image retrieval method should be insensitive to these variations.

The next experiment is designed to explore the capacities of the CTBIR method for resisting the variations of rotation, distortion, noise, scale, hue, luminance, and contrast in images. 100 full color images $I_{q,1}, I_{q,2}, \dots, I_{q,100}$ are used as the query images. These 100 full color images comprise an image set S_q . Besides, this paper employs the rotation, distortion, noise, scale, hue, luminance, and contrast functions in ADOBE PHOTOSHOP 7.0 to process each $I_{q,i}$, and, respectively, generates the variant images $I_{r,i}, I_{d,i}, I_{n,i}, I_{s,i}, I_{h,i}, I_{l,i}$, and $I_{c,i}$. The group of images $I_{\alpha,1}, I_{\alpha,2}, \dots, I_{\alpha,100}$ forms an image set S_α , which is produced by the same function where $\alpha = r, d, n, s, h, l$, and c . Figure 7 shows some images in $S_q, S_r, S_d, S_n, S_s, S_h, S_l$, and S_c , respectively, where i is the image number in $S_q, S_r, S_d, S_n, S_s, S_h, S_l$, and S_c .

In this experiment, all the images in $S_r, S_d, S_n, S_s, S_h, S_l$, and S_c are employed as the database images, and the images in S_q as the query images. For the i th query, $I_{q,i}$ is

Table 3: The results of the five experiment.

	CTBIR	Huang's	SamMatch
ANMRR _{total}	0.20	0.29	0.28
ANMRR _r	0.21	0.92	0.95
ANMRR _d	0.15	0.14	0.44
ANMRR _n	0.35	0.54	0.21
ANMRR _s	0.24	0.23	0.28
ANMRR _h	0.19	0.12	0.11
ANMRR _l	0.24	0.26	0.25
ANMRR _c	0.18	0.16	0.02

used as the query image and the CTBIR, Huang's, and SamMatch methods are applied to retrieve the similar database images for the query image. Table 3 shows the experimental results. The ANMRR_{total} in Table 3 is the accuracy (ANMRR) which the experiment obtains. Furthermore, the ANMRR_r, ANMRR_d, ANMRR_n, ANMRR_s, ANMRR_h, ANMRR_l, and ANMRR_c in Table 3 are the accuracies (ANMRR) obtained by the experiments which take the images, respectively, in S_r , S_d , S_n , S_s , S_h , S_l , and S_c as the database images.

The experimental results show that the color feature is more significant than the texture feature in distinguishing different cartoon images; hence, a bigger r_1 and smaller r_2 should be provided. However, a smaller r_1 and a bigger r_2 should be assigned for recognizing natural images. The experimental results also indicate that the CTBIR method can offer impressive performances in resisting the rotation, translation, distortion, noise, scale, hue, light, and contrast variations, especially for distortion, hue, and contrast variations because it is indifferent to the shape, color, and light variations when describing the texture features of an image.

6. Conclusions

In this paper, a CTBIR method is proposed for image retrieval. The GMRFM feature is used to describe the texture and color information of an image. Based on this feature, the CTBIR method is used to deliver the database images, which are most similar to the given query image, to the user. The GBPD is offered to decide the fittest parameters of r_1 and r_2 , as well. The experimental results show that the CTBIR method is indifferent to the rotation, translation, distortion, noise, scale, hue, light, and contrast variations, especially distortion, hue, and contrast variations. Moreover, the CTBIR method could give a much better performances than those of the Huang's and SamMatch methods. In the future, a subimage retrieval system based on the GMRFM feature will be developed. With a given query image I_Q , the subimage retrieval system is to respond the images I_D from a database so that I_Q is very similar to a region on I_D .

References

- [1] D. Y. M. Chan and I. King, "Genetic algorithm for weights assignment in dissimilarity function for trademark retrieval," in *Proceedings of the 3rd International Conference on Visual Information and Information Systems*, vol. 1614 of *Lecture Notes in Computer Science*, pp. 557–565, 1999.

- [2] R. C. Joshi and S. Tapaswi, "Image similarity: a genetic algorithm based approach," *World Academy of Science, Engineering and Technology*, vol. 27, pp. 327–331, 2007.
- [3] W. J. Kou, *Study on image retrieval and ultrasonic diagnosis of breast tumors*, Ph.D. Dissertation, Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan, 2001.
- [4] B. M. Mehre, M. S. Kankanhalli, and W. F. Lee, "Content-based image retrieval using a composite color-shape approach," *Information Processing & Management*, vol. 34, no. 1, pp. 109–120, 1998.
- [5] X. J. Shen and Z. F. Wang, "Feature selection for image retrieval," *Electronics Letters*, vol. 42, no. 6, pp. 337–338, 2006.
- [6] Z. Stejić, Y. Takama, and K. Hirota, "Genetic algorithm-based relevance feedback for image retrieval using local similarity patterns," *Information Processing and Management*, vol. 39, no. 1, pp. 1–23, 2003.
- [7] K. Vu, K. A. Hua, and W. Tavanapong, "Image retrieval based on regions of interest," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, pp. 1045–1049, 2003.
- [8] B. Yue, *SAR sea ice recognition using texture methods*, M.S. thesis, University Waterloo, Waterloo, Canada, 2001.
- [9] 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.
- [10] Y.-K. Chan and C.-Y. Chen, "An image retrieval system based on the feature of color differences among the edges of objects," *Journal of Computer Science & Technology*, vol. 5, no. 1, pp. 25–29, 2005.
- [11] R. Chellappa and S. Chatterjee, "Classification of textures using Gaussian Markov random fields," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 4, pp. 959–963, 1985.
- [12] F. S. Cohen, Z. Fan, and M. A. Patel, "Classification of rotated and scaled textured images using Gaussian Markov random field models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 2, pp. 192–202, 1991.
- [13] J. R. Smith and S. F. Chang, "Automated image retrieval using color and texture," Tech. Rep. CU/CTR 408-95-14, Columbia University, New York, NY, USA, July 1995.
- [14] J. Han and K.-K. Ma, "Fuzzy color histogram and its use in color image retrieval," *IEEE Transactions on Image Processing*, vol. 11, no. 8, pp. 944–952, 2002.
- [15] W. K. Pratt, *Digital Image Processing*, Wiley-Interscience, New York, NY, USA, 3rd edition, 2001.
- [16] F. Liu and R. W. Picard, "Periodicity, directionality, and randomness: world features for image modeling and retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 722–733, 1996.
- [17] W. Niblack, R. Barber, W. Equitz, et al., "The QBIC project: querying images by content, using color, texture, and shape," in *Storage and Retrieval for Image and Video Databases II*, vol. 1908 of *Proceedings of SPIE*, pp. 173–187, 1993.
- [18] W. Niblack, X. Zhu, J. L. Hafner, et al., "Updates to the QBIC system," in *Storage and Retrieval for Image and Video Databases VI*, vol. 3312 of *Proceedings of SPIE*, pp. 150–161, San Jose, Calif, USA, January 1998.
- [19] P. W. Huang and S. K. Dai, "Design of a two-stage content-based image retrieval system using texture similarity," *Information Processing and Management*, vol. 40, no. 1, pp. 81–96, 2004.
- [20] M.-C. Su and C.-H. Chou, "A modified version of the K-means algorithm with a distance based on cluster symmetry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 674–680, 2001.
- [21] R. Chellappa, "Two-dimensional discrete Gaussian Markov random field models for image processing," in *Progress in Pattern Recognition*, vol. 2, pp. 79–112, Elsevier Science B.V., Amsterdam, The Netherlands, 1985.
- [22] S. M. Choi, J. E. Lee, J. Kim, and M. H. Kim, "Volumetric object reconstruction using the 3D-MRF model-based segmentation," *IEEE Transactions on Medical Imaging*, vol. 16, no. 6, pp. 887–892, 1997.
- [23] R. L. Kashyap and R. Chellappa, "Estimation and choice of neighbors in spatial-interaction models of images," *IEEE Transactions on Information Theory*, vol. 29, no. 1, pp. 60–72, 1983.
- [24] K. K. Seo, *Content-Based Image Retrieval by Combining Genetic Algorithm and Support Vector Machine*, vol. 4669, Springer, Berlin, Germany, 2007.
- [25] L. Ballerini, X. Li, R. B. Fisher, and J. Rees, *A Query-by-Example Content-Based Image Retrieval System of Non-Melanoma Skin Lesions*, vol. 5853 of *Lecture Notes in Computer Science*, Springer, Heidelberg, Germany, 2009.

- [26] C.-C. Lai and Y.-C. Chen, "Color image retrieval based on interactive genetic algorithm," in *Proceedings of the 22nd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (IEA/AIE '09)*, vol. 5579 of *Lecture Notes in Computer Science*, pp. 343–349, Tainan, Taiwan, June 2009.
- [27] K. F. Man, K. S. Tang, and S. Kwong, *Genetic Algorithms: Concepts and Designs*, Advanced Textbooks in Control and Signal Processing, Springer, London, UK, 1999.
- [28] "MEPG-7 Visual Part of Experimentation Model (XM) Version 2.0," MPEG-7 Output Document ISO/MPEG, December 1999.