Expert system for color image retrieval

Hun-Woo Yoo\textsuperscript{a,*}, Han-Soo Park\textsuperscript{b}, Dong-Sik Jang\textsuperscript{b}

\textsuperscript{a}Center for Cognitive Science, Yonsei University, 134 Shinchon-Dong, Seodaemun-Ku, Seoul 120-749, South Korea
\textsuperscript{b}Department of Industrial Systems and Information Engineering, Korea University, Sungbuk-gu Anam-dong 5 Ga 1, Seoul 136-701, South Korea

Abstract

Recently, as Web and various databases contain a large number of images, content-based image retrieval (CBIR) applications are greatly needed. This paper proposes a new image retrieval system using color-spatial information from those applications.

First, this paper suggests two kinds of indexing keys to prune away irrelevant images to a given query image: major colors’ set (MCS) signature related with color information and distribution block signature (DBS) related with spatial information. After successively applying these filters to a large database, we get only small amount of high potential candidates that are somewhat similar to a query image. Finally, we retrieve more similar images from the database by comparing a query image with candidate images through a similarity measuring function associated with the weights. In that procedure, we use a new relevance feedback mechanism. This feedback enhances the retrieval effectiveness by dynamically modulating the weights of color-spatial information. Experiments show that the proposed system is not only efficient but also effective.

Keywords: MCS (major colors’ set) signature; DBS (distribution block signature); QM (quad modeling); CBIR (content-based image retrieval); Relevance feedback

1. Introduction

With the increased emphasis on multimedia applications, the production of image information has resulted in a large volume of images that need to be properly indexed for retrieval in the future. In the earlier days, most retrieval systems used the text-based approach (Joseph & Cardenas, 1988; Roussopolous, Faloutsos, & Sellis, 1988). However, because of the burden of manual indexing and the subjectivity of image contents, the content-based retrieval method that indexes data automatically by computer became popular. This approach retrieves images on the basis of color, texture, shape, or spatial relationship of the objects within the image. For example, systems (Sethi et al., 1998; Smith & Chang, 1996; Swain & Ballard, 1991) retrieve images based on a color. System (Liu & Picard, 1996) retrieves images based on their texture. Systems (Del Bimbo & Pala, 1997; Jain & Vailaya, 1998) retrieve images based on their shape, and systems (Carson, Belongie, Greenspan, & Malick, 2002; Flickner et al., 1995; Ma & Manjunath, 1999; Yoo, Jang, Jung, Park, & Song, 2002) retrieve images based on the combination of color, texture, and shape.

In general, humans feel more sensitive to color than to texture and shape. In some cases, color alone contains sufficient image information for retrieval applications. Also, color-based retrieval has the advantage of achieving fast results because it can save the time that would be added to a retrieval phase when texture and shape features are included. Thus, retrieval systems using color have been popular for a long time.

Early color-based retrieval systems have used the global RGB histogram information such as the histogram difference approach (Yoo et al., 2002), the histogram intersection (Sethi et al., 1998; Smith & Chang, 1996; Swain & Ballard, 1991), and the quadratic histogram comparison (Hafner, Sawhney, Equit, Flickner, & Niblack, 1995). These systems, however, do not capture local color information within the image. For example, consider two flag images as in Fig. 1; one (France) has blue, white, and red colors from left to
right, whereas the other (the Netherlands) has red, white,
and blue from top to bottom. In global color-based retrieval,
these two images are considered to be the same. Hence,
color and its associated spatial information have been
exploited for more accurate retrieval. Related researches
include color coherence vector (Pass, Zabih, & Miller,
1996), color correlogram (Huang, Kumar, Mitra, & Zhu,
1998), and binary color set (Cinque, Levialdi, Olsen, &
Pellicano, 1999; Smith & Chang, 1999).

In the researches of Tan, Ooi, and Yee (2001), color-
spatial techniques are classified into three categories: a
partition-based approach, a signature-based approach, and a
cluster-based approach. The partition-based approach
divides an image into a grid of $m \times n$ cells of equal size,
and compute color similarity between corresponding cells
of two images. Then, by summing similarities in all cells,
final similarity between two images are computed. The
signature-based approach also divides an image in the same
manner. However, it constructs a certain signature by extrac-
ting features from each cell and concatenating those features,
and computes image similarity based on the similarity of
these signatures. The cluster-based approach is based on the
fact that two images are similar if they have large patches of
similar colors at roughly the same locations. For that
purpose, the image is first segmented by color infor-mation
into a few major objects. Retrieval is performed based on the
similarity between objects with respect to color, area of
object, spatial location, etc. According to their researches,
the signature-based approach is generally better than the
partition-based and cluster-based methods in terms of
retrieval effectiveness (which is measured by precision and
recall) and efficiency (which is measured by retrieval time).

Image retrieval can be performed through an interactive
session. Images obtained as the answer to a previous query
are fed back with annotations to improve the quality of the
result. This process is hopefully expected to converge to a
satisfactory solution, although there is no assurance of this.
This relevance feedback mechanism has been exploited by
many researchers. The PicHunter System (Cox, Miller,
Minka, Papathomas, & Yianilos, 2000) used a learning idea
with a Bayesian approach. The information-embedding
framework (Lee, Ma, & Zhang, 1999) attempted to embed
semantic information using a semantic correlation
matrix. System (Li, Chen, & Zhang, 2002) estimated the
probability of similarity between two images based on the
cocurrence frequency. Other system (Rui, Huang,
Ortega, & Mehrota, 1998) used a vector space model
where new query feature vector is generated as a weighted
linear combination of the original feature vector. An
artificial neural network (SOFM) is also used for the
feedback (Laaksonen, Koskela, Laakso, & Oja, 2001).

In this paper, we propose a new system using a signature-
based color-spatial image retrieval method. Color and its
spatial distribution within the image are used for features.
This system, as earlier explained, is based on the
observation that two images with similar color as well as
with similar spatial location associated with that color is
similar with respect to human perception. The color-spatial
algorithm is designed to improve the retrieval efficiency and
effectiveness simultaneously. Fig. 2 illustrates the overview
of the proposed algorithm. During database indexing, an
image is first entered into the database and then, two image
features called the GCS (Global Colors’ Set) signature and
the DBS (Distribution Block Signature) are extracted. The
GCS signature contains all color information in the image
and the DBS denotes spatial distribution of a few major
colors among the GCS signature. This process is repeated
for all images to be used for database indexing. During
retrieval, once a query image is presented, the retrieval
system extracts the MCS (Major Colors’ Set) signature and
DBS of the query. Here, the MCS signature represents the
four major colors of a query. Then, database images, which
do not contain all the MCS of a query, are filtered out (first
filtering). This means that only database images, which have
same major colors with those of a query, are used for further
similarity computation. Next, database images that do not
have a similar spatial location to that of the query are filtered
out (second filtering). This means that only images with
similar spatial distribution are chosen as potential candi-
dates. Finally, among these candidate images, a similarity
measure is used to perform a final similarity computation.
In addition, for more accurate retrieval, a new feedback
mechanism based on QM (quad modeling) is employed.

![Image](image_url)
Our contributions in this paper are fourfold. First, the advantage of signature based method is exploited by describing major color information within a query in MCS signature and, based on the MCS signature, a filtering mechanism for screening out database images with much different colors to those of the query is realized for fast retrieval. Second, the use of spatial information of major colors, which is described in DBS makes it possible for obtaining more accurate retrieval. Third, through the QM (quadratic modeling) representation, system robustness against to translation, rotation, and scale change of a major color region is improved. Also, describing local color distribution in multiple resolutions is achieved. Finally, a new feedback mechanism is proposed in order to incorporate human interventions during query sessions.

This paper is organized as follows. In Section 2, the HSV color model and image size normalization schemes are briefly explained. In Section 3, image features such as the MCS signature, the GCS signature, and the DBS are described. The similarity measure using the QM and relevance feedback is presented in Section 4. Section 5 discusses experimental results in terms of efficiency, effectiveness, and feedback influence. Conclusions are given in Section 6.

2. Color and image size information

We used the HSV (Hue, Saturation, Value) color model in our research. HSV is an intuitive color space in the sense that each component contributes directly to visual perception. Hue is used to distinguish colors (e.g. red, yellow, blue) and to determine the redness or greenness, etc. of the light. Saturation is the measure of percentage of white light that is added to a pure color. For example, red is a highly saturated color, whereas pink is less saturated. Value refers to the perceived light intensity.

Color quantization is useful for reducing the calculation cost and for efficient storage. Furthermore, it provides better performance in an image database application because it can eliminate the detailed color components, which can be considered noises. The human visual system is more sensitive to hue than saturation and value so hue should be quantized more finely than saturation and value. In our experiments, we uniformly quantized HSV space into 18 bins for hue (each bin consisting of a range of 20°), three bins for saturation and three bins for value for lower resolution. By adding four grays, a total of 166 colors (18×3×3 quantized colors + 4 grays) are used to represent the image. For details, readers are requested to refer to the research (Yoo et al., 2002).

In order to retrieve similar images, the query image and all database images should be compared under the same conditions. If the size of a query and the database images is different, a suitable normalization is necessary. For that purpose, we partitioned the image into a grid of 32×32 cells of equal size as in Fig. 3.

Once an image is divided, every image has 1024 cells regardless of image size. Representing with a total of 1024 cells makes it possible to handle an image on the cell-level rather than the pixel-level. In order to obtain color distribution of the image, this approach is very useful. It is not only independent of image size, but also reduces computation cost (fast retrieval). In addition, an approximate color spatial approach is possible by using color distribution based on the cell-level.

3. Image features

For image retrieval, the system should extract image features and compute the degree of similarity between a query and a database image. In this section, image features such as the MCS signature, the GCS signature, and the DBS are explained for a color-spatial retrieval application. These features are used for comparing spatial location as well as color in order to achieve accurate and speedy retrieval. During a query process, the first and second filtering schemes associated with the above features are applied to prune away potential images that are irrelevant to the query.

3.1. MCS (major colors’ set) and GCS (global colors’ set)

If a few colors within an image are prominent, retrieval can be achieved based on these colors. Given two images, they will appear to be similar if both of them have a large amount of similar colors. In general, humans tend to perceive the image by the presence of three major colors (Biederman, 1985). In our research, we selected four major colors to represent the query image. In the experiments on our image database, we found that representing with two colors for the background and two colors for the image center was sufficient to roughly represent the image. Hereafter, we shall refer to the four major colors as the MCS (Major Colors’ Set). The MCS is determined as follows. The query image is partitioned into 32×32 cells of equal size. In each cell, a quantized joint HSV histogram is computed to extract the most frequent (highest peak) bin as a dominant color in the cell. Fig. 4 shows the mosaic-like query image after the corresponding peak color is assigned to all pixels in the cell. Even though the image looks like

![Fig. 3. An image partitioned into a 32×32 grid.](image-url)
a low-resolution version, it contains sufficient color information for image retrieval. Finally, the four major (highest peaks) colors (i.e. MCS) are extracted from the histogram of this image. Fig. 5 shows four binary images constructed by four major colors.

In general, many images tend to have a small object area (i.e. Region of Interest) and a large background area. Therefore, there is a possibility that using a simple histogram method to obtain the MCS can eliminate an important color from the region of interest because of a relatively large background area. In order to handle the problem, extracting the MCS separately from the region of interest and the background is highly desirable. A sunset image of Fig. 6a shows a default object area near the image center and a background area. The important object such as a glaring sun is outside of the center in Fig. 6a. Hence, the MCS extracted from the default area does not represent color content well. Therefore, the system is designed to allow the user to flexibly change the region of interest in the query process. Fig. 6b shows the shift of the ROI to the top-right region by a user so that the system captures user’s intention to represent the MCS. After shifting, we extract two major colors MCb from the background area and two major colors MCc, which are not duplicated from MCb, from the ROI.

In a signature-based color-spatial retrieval technique, the color-spatial information is encoded as a set of color signatures, and a signature file is used to perform speedy and accurate retrieval. In our research, only the hue component in color is used for compact representation and fast retrieval. Human perceives hue more sensitively than saturation and value as already mentioned. The Four hues from the four major colors (i.e. MCS) are represented in bit stream (called color signature). The MCS color signature consists of a one-dimensional array with 19 bits. The first 18 bits are used to represent the four hues of the MCS (note that hue is quantized into 18 distinct values as mentioned earlier) and the final bit is used to represent gray information of the MCS. Initially all 19 bits are set to 0. MCS color signature is obtained by assigning bits associated with the four hues of the MCS to 1. The final bit should be set to 1 only when the MCS contains at least more than one gray.

While the MCS color signature is used in the query side, the global colors’ set (GCS) color signature is used in the database side. The GCS color signature captures global color information in the image. It consists of 19 bits like the MCS. However, unlike the MCS signature that is constructed with only the four hues of the four major colors in $32 \times 32$ cells, the GCS signature is constructed with all hues in $32 \times 32$ cells. Each bit of the GCS signature is assigned to 1 when the corresponding hue exists in the cells. Therefore, an image with various colors may contain many 1s, whereas an image with a small number of colors may contain a small number of 1s (MCS signature does not contain above five 1s since at most the four major hues and one gray are taken). The GCS bit stream, which is extracted from each image in the database is indexed off-line, and used later for pruning away any irrelevant database images in performing image retrieval.

For similarity retrieval, comparing the similarity between a query and all database images is time-consuming. Therefore, most retrieval systems use a filtering process where images less similar to the query are eliminated in advance from similarity computation. In order to do that, the MCS signature of a query and the GCS signature of a database image are compared using a bitwise logical-AND operation. Only database images that must have all the MCS of a query are qualified for further similarity computing. That is, after the operation, images with more than four 1s become candidates for further computing (note that the AND operation is 1 only when two compared bits are both 1s). This first filtering process is described in Fig. 7.

3.2. DBS (distribution block signature)

Candidate database images after the first filtering step have similar color content to that of a query image. Next, the second filtering process is performed to compute further similarity by considering color spatial information. We will call the bit stream used in this step the distribution block signature (DBS). Like the GCS signature, the DBS of database images is also indexed into the database off-line. The DBS is extracted as follows. First, an image is partitioned into $4 \times 4$ blocks of equal size. Each block will contain 64 cells since we partitioned the image into $32 \times 32$
cells. Then, with respect to the first color in the MCS, the block is assigned to 1 when it contains same first color, or assigned to 0 when it does not. That produces a 16 bit stream for first major color. The same process is repeated for the rest of the MCS. That results in a total of 4×16 bit streams. Fig. 8 shows the construction of the DBS using one major color of the MCS, which is identical to a plane color.

In the second filtering, by comparing the DBS of a query with the DBS of candidate images, more similar images to the query are taken. In this step, a bitwise logical-XOR operation is applied since dissimilar color-spatial blocks as well as similar color-spatial blocks are considered at the same time (note that the XOR operation is 0 only when the two compared bits are both 1s or 0s). Fig. 9 shows the second filtering process.

For one major color, the XOR operation produces the sum of all 16 bits ranging from 0 to 16. A result 0 means that the two images have completely the same color-spatial information, whereas 16 means the two images have completely dissimilar color-spatial information. Considering all four major colors, it produces the sum ranging from 0 to 64. In our experiments, a threshold 43 was a good choice for filtering out dissimilar images.

4. Similarity computation

For candidate images after the first and second filtering processes, a final similarity is computed. In order to do that, for each major color of a query image, a 32×32 weight matrix called quad modeling (QM) is computed. On the database side, a 32×32 cell matrix is extracted for the same colors. Image similarity is decided based on the amount of color-spatial overlap, which is computed by comparing each element of the two matrices. In the process, a new relevance feedback mechanism is applied for a more accurate retrieval.

4.1. QM (quad modeling)

For color-spatial image retrieval, the degree of similarity is decided by the amount of a color-spatial overlap between a query and a candidate image. To this end, a new scheme called QM (quad modeling) is presented to capture the color-spatial information of a query. The QM is a 32×32 weight matrix where each element has the degree of importance of a major color. A total of four 32×32 QM matrices are used to represent the four major colors of a query. Similarly, in a database image, four 32×32 cells are employed. This cell image contains only dominant colors such as the second image of Fig. 4. For a color that is identical to a major color of a query, each cell has a 1 when it contains the corresponding color or a 0 when it does not. (As will be explained later, however, each weight of QM matrices of query has fractional value.) Of course, the information of the four 32×32 cells in each database image is extracted and indexed beforehand off-line. Next, the image similarity is obtained by summing the overlaps of the associated two elements (one is in the QM weight matrix and the other is in the cells) in same locations for the four colors.

For a certain major color of a query, QM have characteristics where the weights of elements that have corresponding color and the weights of neighboring elements are set to higher value. Fig. 10 shows a QM matrix construction procedure for one major color. It illustrates how a binary image with one major color is...
partitioned recursively and how the corresponding QM weight matrix is constructed. In the beginning, each element in the matrix is set to 1. Updated weights of QM are obtained in the following way. First, the binary image is divided into four regions (a2). In each region, check if a region includes a major color or not. If it has a major color, the weight increases by a given step size (1 is used in the figure) or if not, the weight is unchanged (b2). Each region is divided into four regions again (for a total of $4 \times 4 = 16$ regions). In each region, check if the region includes the major color or not, and update the weight again. The same process is recursively performed three more times (total five times) to obtain a $32 \times 32$ QM matrix. Here, each element has a different weight according to whether it has the major color or not. In Fig. 10, the status after three recursive processes is shown for explanation convenience. We assume a step size of 1 for every recursive iteration for the sake of simplicity. In fact, however, the amount of the weight change is based on the size of the region. That is, the step size changes continuously (such as $1/256$, $1/64$, $1/16$, $1/4$, and 1) as the process continues. A resulting $32 \times 32$ QM matrix will have an analogue value ranging from 1 to $277/256$ ($=1 + 1/256 + 1/64 + 1/16 + 1/4$).

This matrix represents the color-spatial information of a query image well. It cannot only describe the distribution of color in the image, but also the gradual transition of color. That is, elements near a major color have higher weights, but elements far from it have lower weights. This is different from the cell of a database image where the cell has 0 or 1. Therefore, to some extent, QM representation is inherently robust for translation, rotation, and scale change of a major color region in a query image. In addition, QM makes it possible for the system to exploit multi-resolution image retrieval. As the partition continues, a coarsely assigned weight has a finer weight, thus capturing more detailed color-spatial information.

In general, color around the image center would dominate overall color perception. Humans tend to ignore the color in the margin of an image or consider it to be not important. Thus, for more accurate retrieval, assigning a higher weight to the center region is desirable. We further updated the QM weight matrix according to the spatial location of the image regions. The amount of change is based on the size of regions. Fig. 11 shows three rectangular regions. The background region 3 is three times larger than region 1 and 1.4 times larger than region 2. We compensated QM matrix weights with a 3:1.4:1 ratio in regions 1, 2, and 3, respectively.

### 4.2. Similarity measure

Once four $32 \times 32$ QM matrices of the MCS in a query image are obtained, it is compared with the four $32 \times 32$ cells of a candidate database image. Let the $Q_{MC}(i,j)$ denote the weight of the $(i,j)$th element of the QM matrix for the $k$th major color in a query image, and $D_{MC}(i,j)$ denote the value of the $(i,j)$th cell of the cell matrix for the corresponding major color in a candidate database image.
Suppose that a database image is partitioned into a 3×3 grid to capture the common information among relevant images. By considering all four major colors, an overall similarity of the images is computed as:

\[
S_{MC[i]} = \sum_{i=0}^{31} \sum_{j=0}^{31} Q_{MC[i]}[i][j] \times D_{MC[i]}[i][j]
\]  

Eq. (1) captures the amount of overlap between the two images for a particular major color and its associated spatial location.

By considering all four major colors, an overall similarity is computed as:

\[
S = S_{MC[0]} + S_{MC[1]} + S_{MC[2]} + S_{MC[3]}
\]

\[
= \sum_{k=0}^{3} \sum_{i=0}^{31} \sum_{j=0}^{31} Q_{MC[k]}[i][j] \times D_{MC[k]}[i][j]
\]  

In terms of color-spatial information, Eq. (2) yields a higher value when two images are similar, whereas it yields a lower value when two images are not similar. By using (2), the proposed system presents the top-40 retrieval results to the user in order of decreasing similarity.

4.3. Relevance feedback

Once a query has been formulated, the system returns an initial set of retrieval results. Sometimes, however, the results are not satisfactory for the user since a simple color-spatial approach does not sufficiently capture human perception of image similarity. Therefore, most retrieval systems employ user feedback called relevance feedback. For more accurate retrieval, we incorporated a new relevance feedback method into our system.

Once the images are retrieved, the system asks the user to check whether each result image is similar to the query or not. Then, the query, augmented by information from only similar results is then resubmitted and processed to obtain better retrieval results. This feedback cycle can be repeated until the user is satisfied with the results. In our research, a lower weight should be assigned. Initially, every cell of the feedback matrix is set to 1. As feedback continues, the weight of an important cell remains at 1, but that of other cells is reduced according to the amount of overlap. This update scheme is based on the following equation:

\[
R_{FMC[i]}[i][j] = 1 - (n - \text{Count}) \times \text{(Step\_Size)}
\]

where \(R_{FMC[i]}[i][j]\) denotes the 32×32 feedback matrix and \(n\) denotes the total number of relevant images. Count represents the number of overlaps at the \(i,j\)th cell. Step\_Size denotes the amount of weight change (in the experiment, 0.05 was used).

Once the \(R_{FMC[i]}[i][j]\) matrix is computed in a current feedback iteration, it is convolved by the QM weight matrix \(Q_{MC[i]}[i][j]\) as in the following equation:

\[
Q_{MC[i]}[i][j] = Q_{MC[i]}[i][j] \times R_{FMC[i]}[i][j]\]

Now, using Eq. (2) with the updated QM weight matrix \(Q_{MC[i]}[i][j]\) from (4), we can achieve more accurate retrieval since the QM matrix uses the common information among the relevant images. This process is repeated until the user is satisfied with the results.

The weight of \(Q_{MC[i]}[i][j]\) describes a relative importance of the \(i,j\)th major color at \((i,j)\)th location. The proposed system is also designed to manually assign the relative importance among the MCSs (\(S_{MC[0]}\), \(S_{MC[1]}\), \(S_{MC[2]}\), and \(S_{MC[3]}\)) (see the combo box in Fig. 5). This makes it possible for the user to achieve more flexible retrievals by assigning higher weights to colors of interest. A similarity function of Eq. (2) represents assigning the same weights among major colors. Thus, it is modified by assigning a different weight on each major color into Eq. (5)

\[
S = W_{MC[0]}S_{MC[0]} + W_{MC[1]}S_{MC[1]}
+ W_{MC[2]}S_{MC[2]} + W_{MC[3]}S_{MC[3]}
\]

\[
= \sum_{k=0}^{3} \sum_{i=0}^{31} \sum_{j=0}^{31} W_{MC[k]} \times Q_{MC[k]}[i][j] \times D_{MC[k]}[i][j]
\]

where \(W_{MC[0]}\), \(W_{MC[1]}\), \(W_{MC[2]}\), and \(W_{MC[3]}\) denote color weight on \(S_{MC[0]}\), \(S_{MC[1]}\), \(S_{MC[2]}\), and \(S_{MC[3]}\), respectively.
5. Experiments

In order to evaluate the proposed image retrieval system, all experiments were performed on a Pentium III with 256 MB of main memory and 20 GB of storage. The programs have been implemented in Visual C++. The Graphic User Interface (GUI) is shown in Fig. 13.

In our experiments, 2000 images were chosen, consisting of 15 categories including airplanes, mountains, bears, dolphins, cheetahs, eagles, elephants, horses, scenery, polar bears, roses, sunsets, tigers, zebras, and the rest. Fig. 14 shows example pictures for each category.

We evaluated the system’s efficiency and effectiveness. For efficiency, we derived statistically how many images were expected to be filtered out during the first and second filtering processes. In other words, we derived how many candidate images were used to compute the final similarity retrieval. This criterion represents system’s retrieval time.

For effectiveness, we examined how many relevant images to the query were retrieved. A decision of ‘relevant’ is based on the retrieval results obtained manually by human subjects and assigned by pressing button (captioned Relevant?) under each image (see Fig. 13). The retrieval effectiveness can be defined in terms of precision and recall rates. A precision rate can be defined as the percent of retrieved images similar to the query among the total number of retrieved images. A recall rate is defined as the percent of retrieved images, which are similar to the query among the total number of images similar to the query in the database. The recall and precision rates are computed by

![Display section of query image](image1)

Fig. 13. The GUI (graphic user interface) of the system.

![Display section of retrieved images](image2)

![Analyzing section of query image](image3)

Fig. 14. Example pictures from each category (categories 1–15 in the order from top to bottom and from left to right).
using following equations:

\[
\text{precision} = \frac{N_{\text{correct}}}{N_{\text{correct}} + N_{\text{false}}} \quad (6)
\]

\[
\text{recall} = \frac{N_{\text{correct}}}{N_{\text{total}}} \quad N_{\text{total}} = N_{\text{correct}} + N_{\text{missed}} \quad (7)
\]

where \(N_{\text{total}}\) denotes total number of database images similar to the query and \(N_{\text{correct}}\) denotes the number of retrieved images similar to the query. \(N_{\text{false}}\) is the number of retrieved images dissimilar to the query and \(N_{\text{missed}}\) is the number of database images that are similar to the query but not retrieved. In the experiments, we used the top-40 retrieved images to compute precision and recall. Also, we evaluated the effect of relevance feedback on more accurate retrieval.

5.1. Efficiency

5.1.1. 1st filtering

After extracting the GCS bit stream that consists of 19 bits, we noticed that most of the 2000 experimental images have 3–19 bits with values of 1. Fig. 15 shows the number of images according to the number of bits with values of 1. While image categories such as roses and horses contain many 1s because those images have a variety of background colors, image categories such as airplanes and sunsets contain fewer 1s because those images have a relatively uniform color distribution in both center and background areas. In our database, among the 19 bits, 12.06 bits on average, with a standard deviation of 3.923 have a value of 1. In the case of the MCS bit stream, we noticed that 3.04 bits on average have 1s (four major colors will not always have four hues (i.e. 4 bits) since there will be a chance that some major colors will have the values of same hues). Hence, for databases that have an average of 12.06 GCS bits with values of 1 and an average of 3.04 MCS bits with values of 1, and under the assumption of normal distribution, we can expect the filtering efficiency to be as follows.

The number of different GCS: \(A C_B = 19C_{12}\). The number of different MCS: \(A C_C = 19C_{3}\).

\[
\text{Expected filtering efficiency} = \left( \frac{8C_C}{A C_C} \right) \times \text{record number}
\]

\[
= \left( \frac{12C_3}{19C_3} \right) \times 2000
\]

\[
= (0.227038) \times 2000 \approx 454.08 \quad (8)
\]

where \(\text{record number}\) represents the total number of database images.

From the above, we can filter out about 77.3% of database images. Thus, statistically, only 454 images among 2000 images can be expected to be chosen as candidate images to further compute similarity retrieval.

5.1.2. 2nd filtering

Next, the 454 images remaining after the first filtering are compared in the second filtering process with respect to the DBS. For a certain major color, dissimilarity ranges from 0 to 16. Considering the four major colors, we could obtain dissimilarities between 0 and 64. Intuitively, 64 means that the two compared images are totally different in terms of color distribution, whereas zero means that the two images are identical. In our experiments, we filtered out images with dissimilarity of 43 or higher, which lead us to eliminate roughly 33% images from the candidate images remaining after the first filtering. Thus, we can expect that 304 images among all 2000 images remain for the final similarity retrieval. Statistically, this means an advantage of 84.75% search space reduction after the first and second filterings.

5.2. Effectiveness

As mentioned earlier, for effectiveness, we computed the precision and recall rates. In particular, we computed these according to images categories. Fig. 16 shows the results. Here, a horizontal axis denotes the image category, whereas a vertical axis denotes recall and precision rates associated with those categories. For one example, in a query from category 5 (cheetah images), an average recall of 0.93 and an average precision of 0.33 are achieved. As a whole, average recall and precision are 0.72 and 0.34, respectively. Fig. 16 shows that overall precision and recall plots are in
similar transitions according to category except for category 11 (roses) where the precision surges radically. This is because the top-40 retrieved rose images have similar color-spatial distribution, but other rose images do not. Categories 8 (horse images), 11 (rose images), and 12 (sunset images) have above average precision. This is because the major color area of these images is larger than that of other categories. Thus, it had more discriminating power. Categories 1 (airplane images) and 4 (dolphin images) show low precision because the sea and the sky in their backgrounds have a similar blue color. Categories 7 (elephant) and 9 (scenery) also show low precision because most of the images have a variety of color distribution (that is, relatively small major color regions).

5.3. Relevance feedback

In order to examine the effect of relevance feedback, we performed the proposed feedback mechanism on images with the lowest precision from each category except categories 2 (mountain images) and 9 (scenery images). In our database, images within categories 2 and 9 have strong semantic relation but weak color-spatial relation. Therefore, we thought the feedback on these two categories would affect negatively retrieval performance (or show no effect at all). For fairness, the system is tested by subjects who were not involved in the development of the system. Feedback iteration is performed only once. The results are shown in Fig. 17. Initially, the average precision without feedback was 0.16. However, the average precision after the feedback mechanism rose up to 0.25. This is expected because the proposed mechanism assigned less weight to color-spatial elements in non-relevant retrieved images.

6. Conclusions

In this paper, a new color-spatial image retrieval system was presented. Experiments showed the system pruned away 84.75% of database images by a two-step filtering process and obtained an average recall of 0.72 and an average precision of 0.34. Furthermore even better results could be achieved by exploiting a new feedback mechanism along with the newly presented QM scheme. This scheme partitioned the image into $32 \times 32$ cells recursively, assigned values between 0 and 1 to each cell, and made more flexible retrieval possible, which is robust against translation, rotation, and scale changes.

However, although the performance of the proposed system could be one solution for image retrieval application, it had some limitations to overcome. One of them was that the filtered images in the two filtering processes had no chance to be retrieved under the current scheme. The two-step filtering scheme eliminated database images by imposing a certain threshold for a fast search. In this process, some images similar to the query but not similar enough to satisfy the threshold were eliminated from further computation. We think that in order to resolve the problem some mechanisms that employ a flexible threshold are necessary.

References


