Image Retrieval Based on Dominant Texture Features

Tienwei Tsai  
Department of Computer Science and Engineering  
Tatung University  
Taipei, Taiwan, R.O.C.  
twt@mail.chihlee.edu.tw

Yo-Ping Huang  
Department of Computer Science and Engineering  
Tatung University  
Taipei, Taiwan, R.O.C.  
yphuang@ttu.edu.tw

Te-Wei Chiang  
Department of Information Networking Technology  
Chihlee Institute of Technology  
Taipei County, Taiwan, R.O.C.  
ctw@mail.chihlee.edu.tw

Abstract- This paper presents a novel technique that can be used for fast indexing and retrieval of images based on their dominant texture features. Unlike the existing techniques that use computationally intensive texture features for content-based image retrieval, our proposed features are only derived from the DCT coefficients transformed from the Y-component in YUV color space. The dominant texture feature vector is mainly formed with the fundamental properties of global textures. In addition, to employ the fuzzy cognition concepts, our experimental system allows users to easily adjust weights for each individual feature component. Experimental results show that the proposed feature vector is compact with good retrieval accuracy.

Keywords: Content-based image retrieval, discrete cosine transform, texture feature.

I. INTRODUCTION

With the growth of multimedia application and the spread of Internet, the access of digital images becomes a significant need. Unlike the annotation-based retrieval, the content-based image retrieval (CBIR) has emerged as an important area, which tends to index and retrieve the images according to their low-level features including color, texture, shape, etc. Among these features, texture is an important item of information that humans use in analyzing a scene. It is particularly useful in the analysis of natural environment, as most natural scenes consist of textured surface [1].

Since transform coding has been proven to be a promising technique to achieve image data compression, a number of texture analysis methods have been proposed in transform domain of images [2-5]. In most existing transform coding systems, the block size used to divide an input image is fixed. The Discrete Cosine Transform (DCT) is usually chosen because of its high capability of energy compaction and existence of fast computational algorithm. Therefore, most approaches to DCT-based texture classification are based on block DCT transformation. The commonly used block sizes are 8x8 and 16x16. Note that the image described by independent rectangular blocks can lead to visible distortions known as blocking effects due to coarse quantization of the coefficients. Moreover, because the image statistics may be inhomogeneous and vary from area to area in an image, some approaches varied adaptively the block size to yield a better performance of compressions [6,7]. In this paper, our goal does not focus on the compression rate. Instead of taking into consideration of small area, we emphasize the texture features from a global view of an image.

Extending our earlier work [8-10], we propose a generalized global texture feature for image indexing and retrieval in YUV color space. We derive a feature vector for a general case based on the specific location of DCT coefficients denoting different texture properties of an image. The DCT coefficients are computed over the Y-component of an image to achieve a global image texture feature. Then we regard image textures as three one-dimensional vectors characterized by three directional texture properties: vertical, horizontal and diagonal, each of which is located at some specific area of 2-D DCT coefficients. In addition, DC coefficient represents the average energy of an image, which is an important index of an image. Therefore, our proposed feature vector is formed with four major components: the DC coefficient and three directional texture properties. As the classification rule, we employ the distance function on the basis of the sum of squared differences (SSD) without loss of generality.

The contribution of this work lies in generating the dominant features of global image texture. Another advantage is its low computational complexity since the feature vector is formed by a small number of items. The performance of the proposed approach was demonstrated with several experiments. The results show that the proposed dominant texture feature is best suited for discriminating a large number of textured images with low computational complexity. In addition, to employ the fuzzy cognition concepts, our experimental system allows users to easily adjust weights for each feature component. By such multiple processing and refinement of the weights, the retrieval performance can be further improved.

The rest of this paper is organized as follows. In Section II, the texture feature will be defined and then the similarity measurement with the support of a flexible weighting vector will be given and discussed in Section III. In Section IV, we present the experimental results via various queries. Finally, conclusions are drawn in Section V.

II. FEATURE EXTRACTION

Generally, the CBIR systems follow the paradigm of representing images via a set of feature attributes that
generated from their low-level contents. In this paper, we intend to create a content descriptor (or feature vector) formed by global texture features of an image. For large image database, the system can be built to support efficient and effective accesses to image data with a very low computational load using the proposed method.

A. YUV Color Space

There are some existing color models to describe images, known as color spaces, such as RGB, HSV, HIS, and YUV. In our approach, we choose the YUV color model to represent the information of an image. YUV is used in the PAL system of television broadcasting, which is the standard in much of the world. The YUV model defines a color space in terms of one luminance and two chrominance components, which are created from an original RGB (red, green and blue) source. The weighted values of R, G and B are added together to produce a single Y signal, representing the overall brightness, or luminance, of that spot. The U signal is then created by subtracting the Y from the blue signal of the original RGB, and then scaling; and V by subtracting the Y from the red, and then scaling by a different factor. This can be achieved easily with the following formula:

\[ Y(x, y) = 0.299 R(x, y) + 0.587 G(x, y) + 0.114 B(x, y), \]
\[ U(x, y) = 0.492 (B(x, y) - Y(x, y)), \]
\[ V(x, y) = 0.877 (R(x, y) - Y(x, y)). \]

In our approach, the color image is first transformed to YUV color space. However, the proposed texture feature is based on the DCT coefficients that are calculated only from the Y (luminance) component. The main reason for this decision is that human visual system is more sensitive to Y than to other chrominance components.

B. Discrete Cosine Transform

The Discrete Cosine Transform (DCT) can be applied to the entire image or to subimage of various sizes. As described in previous section, most existing work extracts texture features using block-DCT coefficients. In this section, we will describe our approach in this area focusing on using the DCT coefficients transformed from the entire image to generate the global texture feature. The DCT transform for an \( N \times N \) image represented by pixel values \( f(i, j) \) for \( i, j = 0, 1, \ldots, N-1 \) can be defined as

\[
C(u, v) = \frac{2}{N} \alpha(u) \alpha(v) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i, j) \times \cos\left(\frac{(2i + 1)u\pi}{2N}\right) \cos\left(\frac{(2j + 1)v\pi}{2N}\right),
\]

for \( u, v = 0, 1, \ldots, N-1 \), where \( \alpha(w) = 1/\sqrt{2} \) if \( w = 0 \) and 1 otherwise, \( C(u, v) \) are the DCT coefficients and \( f(i, j) \) are the input pixels. We notice that DCT is a unitary transform, which has the energy preservation property, i.e.,

\[
E = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (f(i, j))^2 = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} (C(u, v))^2,
\]

where \( E \) is the signal energy.

Since the DCT transform is a one-to-one mapping, the overall size of the DCT result obtained is equal in size and dimension to the input image, but can be reduced in virtue of its energy compacting property. For most images, much of the signal energy lies at low frequencies; these appear in the upper left corner of the DCT. The lower right values represent higher frequencies, and are often small - small enough to be neglected with little visible distortion. In our case study of OCR, 84% signal energy appears in the 2.8% DCT coefficients, which corresponds to a saving of 97.2% \([11]\). Therefore, we select the upper left corner of DCT to form the feature vector of an image.

C. Texture Feature

DCT techniques can be applied to extract texture feature from the images due to the following characteristics \([2]\):

- the DC coefficient \((i.e., C(0, 0))\) represents the average energy of the image;
- all the remaining AC coefficients contain frequency information which produces a different pattern of image variation; and
- the coefficients of some regions represent some directional information.

As shown in Fig. 1 \([12]\), the coefficients of most upper region, those of most left region and those of most diagonal region in a DCT block represent some vertical, horizontal, and diagonal edge information, respectively.

Based on this observation, the proposed method attempts to categorize the upper left DCT coefficients of an image into several regions, each of which contains its most representative coefficients of directional texture feature. Note that the DCT coefficients are computed over the Y-component of an image to achieve a global image texture feature. Then we regard image textures as three one-dimensional vectors characterized by three texture properties: vertical, horizontal and diagonal, each of which is located at some specific region in 2-D DCT coefficients. In addition, DC coefficient represents the average energy of an image, which is an important index of an image and sensitive to human eyes. Therefore, it is included in the

![Fig. 1. Basic pattern in a DCT block](image-url)
The proposed feature vector. Fig. 2 shows typical regions of significant texture coefficients used in our method. As such, our proposed feature vector is formed with four major components made by numerous low frequency coefficients: one DC coefficient \( F_1 \) and three directional texture vectors \( F_2, F_3, \) and \( F_4 \). They can be defined as:

\[
F_1 = (C_{00}), \\
F_2 = (C_{01}, C_{02}, C_{03}), \\
F_3 = (C_{10}, C_{12}, C_{13}, C_{11}, C_{14}, C_{15}), \\
F_4 = (C_{01}, C_{02}, C_{03}, C_{04}, C_{05}, C_{06}, C_{07}, C_{08}, C_{09}, C_{10}, C_{11}, C_{12}, C_{13}, C_{14}, C_{15}).
\]

The proposed feature descriptor \( F \) consisting of above components is denoted in vector form:

\[
F = [F_1, F_2, F_3, F_4],
\]

where \( ||F_1|| = 1, ||F_2|| = 9, ||F_3|| = 9, \) and \( ||F_4|| = 11. \)

Note that \( ||F_m|| \) is the number of coefficients in the feature vector \( F_m \) for \( m = 1 \) to \( 4 \). These coefficients are deliberately chosen as they are essential to the grayness and directionality of an image. Moreover, only the low frequency coefficients are chosen because they convey higher energy level in a typical DCT domain. The entire image is summarized into a feature vector \( F \) that describes its most dominant features. As a result, the total number of DCT coefficients in \( F \) is 30, which is determined by numerous experiments. Our experiment showed that more high frequency coefficients gave little improvement on the retrieval performance.

III. SIMILARITY MEASUREMENT

For similarity-based retrieval, descriptors of images in an image database are first constructed and stored in an index file linked to the original images. To decide which image in the image database is the most similar one with the query image, we have to define a measure to indicate the degree of dissimilarity (or distance).

A. Distance Function

The distance between a feature vector \( F_m \) of the query image and that of an image in the database is based on the distance function. To exploit the energy preservation property of DCT as shown in (5), we calculate the distance between two vectors on the basis of the sum of squared differences (SSD). Assume that \( F_{qm} \) and \( F_{xm} \) represent the \( m \)th feature of the query image \( Q \) and an image \( X \) in the database, respectively. Then, the distance between \( F_{qm} \) and \( F_{xm} \) can be defined as

\[
d_m(F_{qm}, F_{xm}) = \sum_{i=0}^{4} (F_{qm}[i] - F_{xm}[i])^2,
\]

where \( i \) is the \( i \)th component of \( F_{qm} \) and \( ||F_{qm}|| = ||F_{xm}|| = k \). Note that the computation cost is highly dependent on the dimension of the vector. Fewer items in a feature vector will lead to a faster matching. Therefore, using low dimensional feature is an important (and challenging) issue in CBIR and it is the very problem we try to tackle in this paper.

B. Weighting Vector

Since several features are used simultaneously, it is necessary to integrate similarity scores resulting from each individual feature. In our case, the total distance can be derived from the following equation:

\[
D(Q, X) = \sum_{m=1}^{4} w_m d_m(F_{qm}, F_{xm}).
\]

Here, \( Q \) and \( X \) are the query image and one of the images in the image database, respectively. \( d_m \) is the distance function defined as (6) and \( w_m \) is the weight of the \( m \)th feature. That is, \( w_1, w_2, w_3, \) and \( w_4 \) indicate significant levels for grayness, vertical texture, horizontal texture and diagonal texture, respectively. In this paper, the weighting vector \( W \) is in the form of \( (w_1, w_2, w_3, w_4) \) for convenience of expression.

From another point of view, each weight can be regarded as the fuzziness of the cognition to the associated feature. Users can emphasize the features that are relatively important based on their feelings or opinions. An important part of our experimental system is the implementation of a friendly GUI that can be used to easily adjust the value of each weight. Following a visual user interface [10], we translate “hard” number into visual information by the way of scroll bar representation. These scroll bars can be used to express users’ perceptions for each individual feature.

IV. EXPERIMENTAL RESULTS

Having constructed the dominant texture feature vector, we evaluated performance on a test image database, which was downloaded from the WBIIS database [13]. It is a general-purpose database including 1000 uncompressed color images in bitmap format. The images are mostly photographic and have various contents, such as natural scenes, animals, insects, building, people, and so on. Since the images are spatial RGB color image, they are first converted into YUV color space, and then performed the DCT transformation over the Y-component.
Fig. 3 shows the main screen of our experimental system. A user may select any image in the database as a query image. Then, the user can give a set of weights to indicate the relatively importance of directional textures based on his/her impressions or opinions. All these parameters are given by the user either typing the values or dragging the scroll bars. The output are listed on the basis of how well they match the query image and ranked in descending order from left to right then from top to bottom. The retrieved results are promising in this query. Note that an image is regarded as a “correct” retrieval if it contains objects similar to the query, as judged subjectively.

The first experiment is designed to examine the effect of the weighting vector. For the first query, we obtain five correct answers (related to butterflies) in the top 10 using the same weight for each feature, i.e., \( W = (1, 1, 1, 1) \), as shown in Fig. 4(b). Seven correct answers are obtained when we use \( (0.5, 1, 1, 0.5) \) instead in the second query, as shown in Fig. 4(c). The user may adjust the weights if he/she is not satisfied with the current retrieval results. Such multiple passes of retrieval can be of great help in a CBIR system, particularly in situations in which the user does not have a clear target in mind.

For comparison, the Bae’s approach [2] is examined in the following experiments. An image of a flower is given as the query example (Fig. 5(a)). We obtain eight and nine correct answers (related to flowers) using the Bae’s approach and our approach, as shown in Fig. 5(b) and 5(c), respectively. Another image of a sunset scene is given for the next query (Fig. 6(a)). Five correct answers are obtained using our approach while only three correct answers are obtained using the Bae’s approach. The results show that our approach leads to more promising precision and quality than Bae’s. It is probably because the color images in the Bae’s approach are required to covert into gray level images by averaging the value of RGB, which does not sufficiently capture color information in the images. The Y component used in our approach does capture human’s perceptions on the luminance to some extent. In summary, the experimental retrieval performance is good, especially when the query images have obvious texture properties, which vindicates the effectiveness of the proposed dominant texture features.

V. CONCLUSIONS AND FUTURE WORK

Many features have been used in CBIR. This paper mainly focuses on the efficient indexing and retrieving of dominant textures as a first step towards an effective CBIR system. To perform image retrieval based on the textures of an image in a global sense, the proposed approach tries to identify and measure texture features that are considered dominant for human perceptions. We also propose a weighting vector to represent the significant level of the perceptual features: grayness and textural directionality.

The texture features used here are derived directly from DCT coefficients transformed from the Y-component. Since the feature vector is formed by a small number of items, it is computationally less expensive than other approaches. The experimental results also show that making use of the proposed
structure of dominant texture descriptor, an effective and fast retrieval of images can be achieved. In addition, retrieval results can be further improved using a set of weights adjustable based on users’ perceptions.

Preliminary results indicate the appropriateness of using dominant texture features in CBIR. It can also be seen that our method may be better suited for the retrieval of landscapes (natural scenes), where colors are relatively constant or textures are relatively obvious (grass has a yellowish-green hue, sky has a blue hue, etc). However, there still exists some work to be done in our approach. For example, the real-world textures can occur at arbitrary spatial resolutions and rotations and they may be subjected to varying illumination conditions. The gray-scale invariant is often important due to uneven illumination. This inspires us to incorporate invariance with respect to one or some of spatial scale, orientation, and gray scale in the future.

**ACKNOWLEDGMENT**

This work is supported by National Science Council, Taiwan, R.O.C. under Grants NSC94-2745-E-036-002-URD and NSC94-2213-E-036-021.

**REFERENCES**


Fig. 5. (a) The query image, (b) retrieved results using Bae’s approach, and (c) retrieved results using our approach with $W=(0.5, 1, 0.5, 0.5)$.

Fig. 6. (a) The query image, (b) retrieved results using Bae’s approach, and (c) retrieved results using our approach with $W=(0.5, 1, 0.5, 1)$. 