Dominant Feature Extraction in Block-DCT Domain

Tienwei Tsai, Yo-Ping Huang, Senior Member, IEEE, and Te-Wei Chiang

Abstract—Automatically retrieving images through their low-level visual features has become one of the challenging areas of research recently. Among those distinguishing features, the texture features are one of the main themes in content-based image retrieval (CBIR). In this paper, we propose a novel technique to extract dominant features of images in block-DCT domain. The image is first converted to YUV color space and divided into four subblocks. The Y-component in each subblock is then transformed into DCT coefficients, some regions of which characterize different directional texture feature of that subblock. The directional textures in all subblocks are concatenated together as a single feature vector and used for indexing and retrieval of images. The experimental results show that using proper size of block-DCT to emphasize the regional properties of an image while maintaining its global view performs well in CBIR.

I. INTRODUCTION

WITH advances in the Internet and new digital image sensor technologies, the volume of digital images produced by scientific, educational, medical, industrial, and other applications has increased dramatically. Thus, there is a need to find an efficient and effective technique that can be applied to index and retrieve the desired images. Along this line, textual specification of images is a simple and convenient way to express the contents of images. However, it needs a commonly accepted correspondence to the images, which is difficult to achieve, even in culturally homogeneous contexts. Moreover, textual queries are only suited for queries on image semantic since semantic (abstract) concepts are difficult to express using only keywords. Rather than relying on manual indexing or text descriptions, a content-based image retrieval (CBIR) system uses low-level visual contents that can be extracted from the image files, for indexing and retrieving a database of images. Shape, texture, and color are three main groups of features that are being used in CBIR systems. Among those features, texture is usually characterized by the spatial variation, directionality, and coarseness in the image. It is particularly useful in the analysis of natural environment, as most natural scenes consist of textured surface [1].

Developed by Ahmed et al. [2], the Discrete Cosine Transform (DCT) is a technique for separating the image into parts (or spectral sub-bands) of differing importance (with respect to the image's visual quality). The DCT has been proved successful at de-correlating and concentrating the energy of image data. It has brought on the proliferation of visual data stored in the JPEG and MPEG compressed formats. This has made some significance influence on the image retrieval research and application [3]. For example, a number of texture analysis methods have been proposed in DCT domain [3]–[8]. In some approaches [4], [5], an image was regarded as a whole block so as to reduce the computation cost. However, such image-wide DCT has advantage of losing the locality information. Some approaches, proposed in [3], [6]–[8], derived texture features from small size of DCT blocks. For example, 4x4 DCT blocks were used in [6] and 8x8 DCT blocks were used in [3], [7]. As a result, too many decomposition procedures increase the complexity and computational requirements.

Being an elementary process, the feature extraction will be invoked very frequently; therefore, it should be time-efficient and accurate. To reduce the processing time and to still retain the locality information to some extent, we propose a novel technique to extract dominant features of images using proper block size of block DCT (BDCT), whose contribution to the texture feature is more than the entire block model. To perform a block-based DCT, an image is divided into nonoverlapping \( S \times 2 \) subblocks. The research in [9] shows that the BDCT with subblock number \( 2 \times 2 \) performs well in terms of precision rate. Therefore, the image in our approach is divided into four subblocks.

In our approach, the image is first converted to YUV color space and divided into four subblocks. The Y-component in each subblock is then transformed into DCT coefficients, some regions of which characterize different directional texture features of that subblock. The average grayness and directional textures in all subblocks are concatenated together as a single feature vector and used for indexing and retrieval of images. The experimental results show that using proper size of BDCT to emphasize the regional properties of an image while maintaining its global view performs well in CBIR.

The remainder of this paper is organized as follows. Some theoretical background about dominant feature extraction is
presented in Section 2 followed by similarity measurement in Section 3. Section 4 presents experimental results. Finally, conclusions are drawn in Section 5.

II. TEXTURE FEATURE EXTRACTION

The proposed method attempts to categorize the upper left DCT coefficients of a subblock into several regions, each of which contains its most representative coefficients of directional texture feature: vertical, horizontal and diagonal. 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 also included in the proposed feature vector. The procedures of feature extraction are described in the following subsections.

A. YUV Color Space

For the DCT transform, we have to convert an RGB image into gray-level image as DCT works on a single component of color space. A gray-level digital image can be defined to be a function of two variables, \( f(x, y) \), where \( x \) and \( y \) are spatial coordinates, and the amplitude \( f \) at a given pair of coordinates is called the intensity of the image at that point. There are some existing color models such as RGB, HSV, HSI, and YUV. Originally used for PAL (European "standard") analog video, YUV is based on the CIE Y primary, and also chrominance. The Y primary was specifically designed to follow the luminous efficiency function of human eyes. \( U \) and \( V \) provide color information and are "color difference" signals of blue minus luminance (i.e., B-Y) and red minus luminance (i.e., R-Y). The following equations are used to convert from RGB to YUV spaces:

\[
Y(x, y) = 0.299R(x, y) + 0.587G(x, y) + 0.114B(x, y),
\]

\[
U(x, y) = 0.492(B(x, y) - Y(x, y)),
\]

\[
V(x, y) = 0.877(R(x, y) - Y(x, y)).
\]

Psycho-perceptual studies have shown that the human brain perceives images largely based on their luminance value, and only secondarily based on their color information. Therefore, the DCT transformation is performed on the Y component of each subblock.

B. Block-based DCT Transformation

DCT uses the orthogonal real basis vectors whose components are cosines. It can be applied to the entire image or to the subimage of various sizes. For spatial localization, we use the block-based DCT transformation. Each image is divided into \( N \times M \) sized subblocks. The DCT coefficients for an \( N \times M \) image are generated on a pixel by pixel basis, which give the nature of textual energy for each pixel. Note that the DCT coefficients are computed over the Y-component of a subblock to achieve its texture properties. The equation used for the DCT calculation for each pixel is derived from the following equation:

\[
F(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \times \cos \left( \frac{(2x + 1)u\pi}{2N} \right) \times \cos \left( \frac{(2y + 1)v\pi}{2M} \right),
\]

where \( f(x, y) \) is the pixel value at the \( (x, y) \) coordinate position in the image, \( F(u, v) \) is DCT domain representation of \( f(x, y) \), where \( u \) and \( v \) represent vertical and horizontal frequencies, respectively. \( u \) and \( x \) have values from 0 to \( N-1 \), \( v \) and \( y \) have values from 0 to \( M-1 \), and

\[
\alpha(u) = \sqrt{\frac{1}{N}} \quad \alpha(v) = \sqrt{\frac{1}{M}} \quad \text{for } u, v = 0, \text{ and}
\]

\[
\alpha(u) = \sqrt{\frac{2}{N}} \quad \alpha(v) = \sqrt{\frac{2}{M}} \quad \text{for } u, v \neq 0.
\]

It is known that DCT is a reversible transform which obey the energy preservation theorem - total energy in pre-transform domain is equal to total energy in post transform domain. That is:

\[
E = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (f(x, y))^2 = \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} (F(u, v))^2,
\]

where \( E \) is the signal energy. Additionally, DCT is a good example of energy compacting transform. An energy compacting transformation for images is to transform an image vector to another vector such that most of the energy of the vector, on the average, in the new representation is concentrated in a few coefficients. For most images, much of the signal energy lies at low frequencies; these appear in the upper left corner of the DCT. Therefore, only the coefficients with small \( u \) and \( v \) are considered in our approach.

C. Feature Extraction

One of the most important issues in a CBIR system is the feature extraction process, where the visual content of the images is mapped to a certain metric space, usually reducing dimensionality in the process. The proposed feature vector is
formed with four major components made by numerous low frequency coefficients spread in each subblock: one DC coefficient \( F_1 \) and three directional texture vectors \( F_2, F_3, \) and \( F_4 \). Not all the DCT coefficients contain useful information. Only the coefficients that appear in the upper left corner in each subblock are considered. Fig. 1 shows the regions of significant texture coefficients in each subblock \( B_{ij} \) (\( i = 1, 2 \) and \( j = 1, 2 \)). The features in one subblock \( B_{ij} \) can be defined as:

\[
A_y = (C_{y0}), \\
V_y = (C_{y0}, C_{y2}, C_{y4}, C_{y3}), \\
H_y = (C_{y0}, C_{y0}, C_{y4}, C_{y3}), \text{ and} \\
D_y = (C_{y1}, C_{y2}, C_{y4}, C_{y3}).
\]

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

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

where

\[
F_1 = [A_{11}, A_{12}, A_{21}, A_{22}], \\
F_2 = [V_{11}, V_{12}, V_{21}, V_{22}], \\
F_3 = [H_{11}, H_{12}, H_{21}, H_{22}], \text{ and} \\
F_4 = [D_{11}, D_{12}, D_{21}, D_{22}].
\]

Note that \( F_1, F_2, F_3 \) and \( F_4 \) represent the average grayness, vertical texture, horizontal texture and diagonal texture of a specific image subblock, respectively. These coefficients are deliberately chosen as they are essential to the grayness and directionality of an image. Moreover, only the low frequency coefficients of the subblock are chosen because they convey higher energy level in a typical DCT domain. As a result, the total number of DCT coefficients in \( F \) is 64, which is determined by numerous experiments. In other words, the image feature vector is obtained by a 64-dimensional space of the low frequency coefficients aggregated from each subblock. Our experiments show that more high-frequency coefficients gave little improvement on the retrieval performance, as they are comparatively small, and therefore negligible in most cases.

#### III. SIMILARITY MEASUREMENT

Once the feature vector is extracted, the next step on the process of image retrieval is to define similarity relation between pairs of images that expresses how similar they are. Similarity (or dissimilarity) relation is based on a Euclidean-like distance function, which defines the distance between a query image and images in the database.

##### A. Distance Function

The overall distance is the weighted sum of the distances, each of which is derived from an individual feature. As described above, each feature is represented by a feature vector in form of \( F_m \) for \( m = 1 \) to 4. To exploit the energy preservation property of DCT as shown in (5), we calculate the distance between two feature vectors on the basis of the sum of squared differences (SSD). Let \( Q \) be a query image and \( X \) be a database image. Assume the distance \( d_m \) is the corresponding distance measure of the \( m \)th feature; \( F_m^Q \) and \( F_m^X \) represent the \( m \)th feature of \( Q \) and \( X \), respectively. Then,

\[
d_m(F_m^Q, F_m^X) = \sum_{i=0}^{4} (F_m^Q[i] - F_m^X[i])^2.
\]

Here, \( i \) is the \( i \)th coefficient of the \( m \)th feature and \( \|F_m^Q\| = \|F_m^X\| = k \). It is observed that the performance of retrieval critically relies on the distance function used and the dimension of the feature vector. Fewer items in a feature vector will lead to a faster matching. As only the low frequency DCT coefficients are considered, the whole process of similarity measurement is performed on a reduced number of items. This is one of the main objectives to extract the dominant features from the images in our approach.

##### B. Weighting Vector

Combining the distances from several features can be done in a number of ways. In our approach, a friendly user interface is employed for user to input weights, each of which corresponds to the relative importance of an individual feature. Thus, the overall distance can be derived from the following equation:

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

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 \( w_m \in R \) 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. The final ranking of the images that are returned by the system is affected by the weights that the user has assigned to the features. If the user is not satisfied with the result, he/she can adjust the weights and then perform a next query.

#### IV. EXPERIMENTAL RESULTS

For testing the performance of the proposed method, an experimental CBIR system has been implemented with a general-purpose image database including 1000 color images, which was downloaded from the WBIIS database [10]. The images are mostly photographic and have various contents, such as natural scenes, animals, insects, building, people, and so on. In our approach, these images are first converted into YUV color space, divided into four subblocks, and then performed the DCT transformation over the Y-component for
each subblock.

Fig. 2 is the main screen of the CBIR system. Several queries are carried out for the evaluation of our approach. The user can submit a query through a sample image. The proposed system will output 10 images according to their similarity scores to the query image, ranking in descending order from left to right then from top to bottom. For each query example, we manually examine the precision of the query results. The relevance of image semantic depends on the point-of-view of the user. A retrieved image is considered a match if it belongs to the same category of the query image. Since users may have different correspondence to the same image, we have asked the users joining this project to get the evaluation that most of them agree to.

For expressing users’ perceptions on each individual feature, a weighting vector $W$ in form of $(w_1, w_2, w_3, w_4)$ is used to indicate significant levels for grayness, vertical texture, horizontal texture, and diagonal texture, respectively. Since the system allows the user to express his/her personal view of perceptual texture properties, and thus be possibly biased so as to perform retrieval accordingly, multiple passes of retrieval is provided in the system. The user can adjust the weights according to images that have been answered in a previous query, or select one of them as a query example to perform the next query.

For the first query, an image of space beyond the atmosphere is used as the query example (see Fig. 3(a)). We obtain 7 matches (related to the same category) in the top 10 retrieved images which are closest to the query using $W = (0.5, 0.5, 1, 1)$, as shown in Fig. 3(b). 7 matched answers (related to celestial bodies) are obtained as well for $W = (0.5, 1, 1, 1)$ in the second query, as shown in Fig. 4(b). Note that the system permits the user to submit a coarse initial query and continuously refine it using different weights after viewing the retrieved images. Such multiple passes of retrieval is of great help in a CBIR system, particularly in situations in which the user does not have a clear target in mind.

An image of a dusk scene is given in the third query. We obtain eight matched answers (related to dusk) using the weighting vector $(0.5, 1, 1, 1)$, as shown in Fig. 5(b). In Fig. 6(b), we have ten matched answers (related to flowers) using the weighting vector $(0.5, 1, 1, 1)$ in the fourth query. It is observed that queries will lead to higher retrieval performance when the query images have obvious texture properties, which vindicates the effectiveness of the proposed dominant texture features. However, evaluation of a certain approach’s performance in CBIR is often difficult because up to this date, there is no agreed measuring criterions and benchmark testing data set to compare different methods. Generally, our proposed system is fast and its performance results are reasonably close to human perception.

V. CONCLUSION AND FUTURE WORK

CBIR researchers generally acknowledge that semantic retrieval remains impossible. The usual approach is to attempt to characterize the image using a small set of low-level features. These features are often computed globally, and contain no spatial information. To emphasize the regional
properties of an image while maintaining its global view, we propose an efficient approach to retrieve dominant textures in BDCT domain as a first step towards an effective CBIR system. 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, where the user can interact with the system to improve the retrieval performance. Our experimental system shows that the use of dominant features, coupled with a flexible weighting vector, performs well in CBIR.

Many texture features have been used in CBIR. The texture features used here are derived directly from BDCT 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. Preliminary results also indicate the appropriateness of using dominant texture features in CBIR. However, the performance is often poorer as the features from different subblocks may be correlated. Furthermore, the number of subblocks and the boundaries for each subband are empirical values. Work is still in progress to test the retrieval with various block sizes. In practice, the low-level specialized measures of a texture are not effective in some cases, and not every user is sensitive to a particular feature. To this end multiple features are needed, either extracted from images or attached to them. We will introduce other appropriate and useful features for the retrieval task in the future.

REFERENCES

Fig. 5. (a) The query image, and (b) retrieved images: 8 matches out of 10 with $W = (0.5, 1, 1, 1)$.

Fig. 6. (a) The query image, and (b) retrieved images: 10 matches out of 10 with $W = (0.5, 1, 1, 1)$. 